

MINISTRY OF SCIENCE AND EDUCATION OF THE REPUBLIC OF  
KAZAKHSTAN

Non-Profit Joint Stock Company  
ALMATY UNIVERSITY OF POWER ENGINEERING AND  
TELECOMMUNICATIONS

Department Telecommunication system and networks

«Admitted»

Head of the Department Baykenov A.S.  
d.t.s., professor

(Surname and initials, degree, rank)

«      » 20    y.  
(sign)

DIPLOMA PROJECT

Theme: Analysis of the Blockchain technology construction principles,  
algorithms and data structures

Specialty: 5B071900 – Radio engineering electronics and telecommunications

Implemented by: Sharapov N.A. ICTE-14-9  
(Student's surname and initials) group

Scientific Supervisor: Panchenko S.V., M.S., senior lecturer  
(Surname and initials, degree, rank)  
«31» 05 2018 y.  
(sign)

Advisors:  
of Economy section: Tuzelbayev B.I., PhD, associate professor  
(Surname and initials, degree, rank)  
«25» 05 2018 y.  
(sign)

of Life activity safety section: Beginbetova A.S., PhD, senior lecturer  
(Surname and initials, degree, rank)  
«25» 05 2018 y.  
(sign)

of Computer Science section: Panchenko S.V., M.S., senior lecturer  
(Surname and initials, degree, rank)  
«31» 05 2018 y.  
(sign)

Standards compliance controller: Panchenko S.V., M.S., senior lecturer  
(Surname and initials, degree, rank)  
«31» 05 2018 y.  
(sign)

Reviewer: Vassin V.V., M.S., CTO of KVINT LLP  
(Surname and initials, degree, rank)  
«05» 06 2018 y.  
(sign)

Almaty 2018 y.

MINISTRY OF SCIENCE AND EDUCATION OF THE REPUBLIC OF  
KAZAKHSTAN

Non-Profit Joint Stock Company  
ALMATY UNIVERSITY OF POWER ENGINEERING AND  
TELECOMMUNICATIONS

Institute of Space Engineering and Telecommunications (ISET)  
Specialty: 5B071900 – Radio engineering electronics and telecommunications  
Department: Telecommunication systems and networks

ASSIGNMENT

For diploma project implementation

Student: Sharapov Nikita Alexandrovich  
(name, patronymic and surname)

Theme: Analysis of the blockchain technology construction principles algorithms and data structures

Approved by Rector order № 155 of « 23 » 10 20 17 y.

Deadline of completed project: « 25 » 05 20 18 y.

Initial data for project, required parameters of designing result, object initial data:

The prerequisites for this particular project is development of blockchain technology, which gained extremely high popularity in the field of data transmission and storage. It is proposed to develop blockchain application for World Anti-Doping Agency.

List of questions for development in diploma project or brief content:

The main aspect of this project was the development of an application for WADA based on Hyperledger Sawtooth framework. Also in this project were considered the structure of the blocks with all its components: hash functions, blocks, transactions, encryption keys, hard forks and soft forks. Calculations had been made for the reliability of the system and the sizes of the blocks. In the life safety activity section were calculated working conditions for application development. In the economic section were calculated the costs for the purchase and installation of equipment, economic efficiency.

List of illustrations (with exact specifying of mandatory drawing):

Examples of inputs and SHA-256 digest values

Example transaction

A simple network maintaining a copy of a ledger across nodes

Example of a Merkle Tree

Hyperledger Sawtooth structure

Hyperledger Sawtooth transaction process

Configuring variables and creating new key-pairs

Adding table rows and buttons

Interface of the blockchain application

Recommended main references:

Cryptocurrency Technologies: A Comprehensive Introduction,  
Princeton University Press, 2016

Bakre A., Patil N., Gupta S.: Implementing Decentralized Digital  
Identity using Blockchain. - 2017

Hell P., Kelsey J., Shook J.: Cryptocurrency Smart Contracts for  
Distributed Consensus of Public Randomness. 2017.

Polyzos G., Fotiou N.: Blockchain-assisted Information  
Distribution for the Internet of Things, 2017

Project adviser with corresponding sections specifying:

Section	Advisor	Dates	Sign
Life activity safety	Begimbetova A.S.	11.03.2018-25.05.2018	268
Economy section	Tuzelbayer B.T.	16.04.2018-25.05.2018	
Computer science	Panchenko S.V.	28.04.2018-31.05.2018	
Technical part	Panchenko S.V.	18.04.2018-31.05.2018	
Standards compliance	Panchenko S.V.	03.05.2018-31.05.2018	

# SCHEDULE

## of diploma project implementation

No	Sections, list of developing questions	Dates of bringing to Scientific Supervisor	Notes
1.	Introduction	28.11.17 - 02.12.17	Done
2.	Blockchain structure	04.12.17 - 20.12.17	Done
3.	Basic cryptocurrencies	22.12.17 - 25.12.17	Done
4.	Hyperledger technology	27.12.17 - 02.01.18	Done
5.	Purpose and goals of the project	05.01.18 - 06.01.18	Done
6.	Justification of Hyperledger Sawtooth	07.01.18 - 10.01.18	Done
7.	Structure of the framework	13.01.18 - 16.01.18	Done
8.	Network architecture	18.01.18 - 25.01.18	Done
9.	Installation process	03.02.18 - 18.02.18	Done
10.	Calculation of network components	19.02.18 - 09.03.18	Done
11.	Analysis of working conditions	10.03.18 - 15.03.18	Done
12.	Calculation of the natural lightning of the room	17.03.18 - 23.03.18	Done
13.	Calculation of the artificial lightning of the room	25.03.18 - 27.03.18	Done
14.	Technical and economic justification for the realization of blockchain application	29.03.18 - 04.04.18	Done
15.	Calculation of investment costs	09.04.18 - 15.04.18	Done
16.	Calculation of annual operation costs	17.04.18 - 20.04.18	Done
17.	Calculation of income	21.04.18 - 24.04.18	Done
18.	Calculation of economic efficiency	04.05.18 - 07.05.18	Done
19.	Conclusion	12.05.18 - 15.05.18	Done

Assignment issue date « 12 » 01 2018 y.

Head of Department: \_\_\_\_\_

Baykenov A.S.

(Surname and initials)

Scientific Supervisor: \_\_\_\_\_

Panchenko S.V.

(Surname and initials)

Assignment submitted for implementation: \_\_\_\_\_

Sharapov N.A.

(Surname and initials)

## **Аңдатпа**

Бұл дипломдық жобада блокчейн технологиясы деректерді сақтаудың жаңа әдісі ретінде қарастырылды, сонымен қатар Hyperledger Sawtooth негізіндегі фреймворкта Дүниежүзілік Допингке Қарсы Агенттіктің (ДДҚА) үшін блокчейн талдамасы жүзеге асты. Ақпаратты берудің, сақтаудың және алмасудың орталықсыздандырылған әдістері жұмысқа пайдаланылды. Клиент-сервер бөлігінің техникалық есептемелері жасалды.

Тіршілік қауіпсіздікті қамтамасыз ету бөлімінде барлық реттеуші талаптарға жауап беретін жұмыс бөлмесін ұйымдастыру мәселесі қарастырылады. Экономикалық бөлімінде жобаны іске асыру шығындары есептелді.

## **Аннотация**

В данном дипломном проекте была рассмотрена технология блокчейн как новый метод хранения данных, а также осуществлена разработка блокчейн приложения для Всемирного Анти-Допингового Агентства (ВАДА) на базе фреймворка Hyperledger Sawtooth. Для работы были использованы децентрализованные методы передачи, хранения и обмена информации. Произведены технические расчеты клиент-серверной части.

В разделе безопасности жизнедеятельности рассмотрены вопрос организации рабочего помещения, соответствующего всем нормативным требованиям. В экономической части просчитаны затраты на реализацию проекта.

## **Annotation**

In this diploma project, the blockchain technology was considered as a new method of data storage, and the development of a blocking application for the World Anti-Doping Agency (WADA), based on the Hyperledger Sawtooth framework. Decentralized methods of transferring, storing and exchanging information were used at work. Technical calculations of the client-server part were made.

In the section of life safety, the question of the organization of a working area that meets all regulatory requirements were considered. In the economic part, the project implementation costs have been calculated.

## Content

Introduction .....	7
1 Theoretical part .....	9
1.1 Blockchain structure.....	9
1.2 Basic cryptocurrencies .....	15
1.3 Hyperledger technology .....	20
2 Practical realization of the project.....	27
2.1 Purpose and goals of the project .....	27
2.2 Justification of Hyperledger Sawtooth.....	27
2.3 Structure of the framework .....	30
2.4 Network architecture .....	32
2.5 Installation process .....	39
2.6 Calculation of network components.....	44
3 Life activity safety section .....	54
3.1 Analysis of working conditions .....	54
3.2 Calculation of the natural lightning of the room.....	58
3.3 Calculation of the artificial lightning of the room .....	60
4 Economical part.....	63
4.1 Technical and economic justification for the realization of blockchain application .....	63
4.2 Calculation of investment costs .....	64
4.3 Calculation of annual operation costs .....	67
4.4 Calculation of income .....	70
4.5 Calculation of economic efficiency .....	70
Conclusion.....	72
List of abbreviations .....	73
List of references .....	74
Appendix A. Listing of app.js program .....	76
Appendix B. Listing of handlers.js program .....	77
Appendix C. Listing of index.html file .....	78
Appendix D. Anti-plagiarism certificate.....	
Appendix E. Electronic version of the diploma work and demonstration materials (CD-R).....	
Appendix F. Handouts (A4 format – 13 pages) .....	

## Introduction

The blockchain technology that is established recently has received a large number of ledgers. The blockchain acts as an irrevocable account holder so that transactions can be conducted in a decentralized manner. Blockchain-based applications are growing, financial services, reputation systems and the Internet of Things (IOT). However, blockchain technologies faces many challenges such as scalability and security issues waiting to end. This article introduced a comprehensive review of blockchain technology. We first provide an overview of the blocking architecture and compare the specific merging algorithms used in different blockchains. In addition, technical challenges and recent developments are briefly outlined. We also introduced the future of blocking.

Blockchain technologies approach (ie without a central database) is irreversible with digital laser systems, which usually focus on its most basic level, and they enable the user community to implement social labeling records on the community, such as not being able to change any transactions after publication once.

In 2008, the idea at that time was to collect blockchains' many other innovative methods and computer technologies that produced modern cryptocurrencies: electronic currency, electronic encryption protection. The first method based on blocking is Bitcoin. In the novel system of monetary inhibitors in their cumulative value, this information is not a value added to digital wallets - electronic devices (or software) that allow one's electronic transactions.

This wallet is a public record used to transfer prices from one business wallet to another, allowing all network participants to independently verify the validity and accuracy of transactions. Each participant can keep a complete record of all records, so the network can easily change the record or attempt to obtain an exchange. Since a large number of news articles and videos describe blockchain's "magic", this article describes the magic method (ie, how the blocking system works).

Arthur C. Clarke once wrote: "Any advanced enough technology cannot be separated from magic" [1]. The blockchain technique is applicable to the case of emerging applications representing Clarke. There is a high level of use of blockkain, but this technique is not fully understood. This is not a magic; it will not solve all these problems. Like all new technologies, they want to implement each field in various imaginable ways.

This document attempts to improve the understanding of technology so that it can be effectively implemented into World Anti-Doping Agency system.

Some existing block-token technologies will focus on storing wealth, while other smart contracts (software deployed on block selection and software implemented by large-scale computers) have a platform. The new blockhchain technology is being used to enable new usage and increase the efficiency of existing systems. Some implementations of the blockchain cannot be used without permission, so anyone can read and write them. Other embodiments restrict the

participation of specific people or companies, allow slight controls and can be managed by a central entity.

Understanding this information allows the organization to understand what applies to its needs. Although there are many fluctuations in the blockchain system and the rapid development of new technologies, most obstructers use some common concepts. Each transaction has a record of one or more addresses and is digitally signed.

In blockchain, each block is a group of transactions. All transactions in the blockchain are combined with the cryptographic hash of the previous block. Finally, for the current block's title, the new hash must be used in the block data and the next block. Sometimes, by adding a hash of the previous block, each block is blocked in the chain of the current block.

Each technology used in the blockchain system uses the mature concepts of the existing concepts, and is summarized and bundled, which can solve the previous problems. This document examines blockchain's basic knowledge of how to use the technology and what happens if network participants agree that they agree to the transaction and what the law allows

In addition, this diploma work searches for specific blockchain applications and examples when considering using the blockchain system. The use of blockchain technology does not come overnight, and there are some problems, how to deal with malicious users, how to implement control and what blockchain control is restricted. In other words, Blockchain technology is an important concept and will become the basis of many new solutions.

Purpose of the project is blockchain application development on the Hyperledger Sawtooth platform for WADA.

As purpose of the project is clear it is necessary to make list of tasks of the project.

1. Get acquainted with the blockchain technology and existing platforms for building blockchain infrastructure;
2. Understand how the proposed system can be implemented on blockchain
3. Technical embodiment of the idea;
4. Feasibility of creating an application;

# **1 Theoretical part**

## **1.1 Blockchain structure**

Blockchain programs are able to seem complicated; however, they are able to be quickly understood by examining the technologies of each component separately. At a significant level, monetary ideas (such as ledgers) are actually used along with computer science devices (linked prospect lists, distributed networks) as well as encrypted primitives (digital signatures, public/private keys) hybrid blockchain financial concepts.

A crucial aspect of blockchain technology is actually the usage of cryptographic hash functions for a lot of operations, like the contents of hash blocks.

Enter almost any size (for example, a file, some text, or an image), the relative output of the aim (called the message digest, or just digest) to measure the hash method. Even if the smallest change in the input (for example, a single bit) results in mining different outputs, it will show a simple example. The hash algorithm is designed to be unidirectional (also known as preimage resistant): computationally finds any negative input of any pre-specified output map.

In theory, technology is applicable to any type of activity where there is a risk of fraud, mistrust or errors in data transmission.

Let us consider the most promising and effective ways of using blockage.

Method one is Network Administration. Blockchain in this case plays the role of an invulnerable keystore and lists of users who have the right to access any data - servers, terminals, ATM network.

The technology protects against hacker attacks, server errors, hacking of networks and removes the problem of "single administrator".

Method two is storage of digital certificates. Cryptography reliably protects information from unauthorized reading, modification, distribution. Since certificates are stored not on servers, but on the network, it is impossible to obtain illegal access to them, as well as to intercept user passwords [2].

Method three is Proof of ownership. Confirmation and transfer of property rights will become simple, almost instantaneous and safe operations, if we apply block-technology to this sphere.

It is enough for a person who has access to his block to make new information in the block, and information about the ownership right will be distributed throughout the system [2].

Method four is create a DNS system. With the help of the blockbuster, the distribution of names in the domain networks will become absolutely safe. No DDoS attacks will no longer be frightening for ordinary citizens, nor for financial or government organizations.

Method five is identification and confirmation of access rights. Already, some advanced companies are using blocking equipment to identify their customers, employees and system users. The application of the "chain of blocks" is much

cheaper and more efficient than any other methods of protecting data and confirming access rights.

If a particular output is known, unless the hash algorithm produces the desired result, unless you can find an input (called another resistant), the hash function is designed to be phase-tolerant than input: two or more inputs cannot produce the same output

1.1.1 Hashes. These algorithms can be used in blockchain technology hashing algorithm SHA to 256 bits (SHA 256) is the size of the output. Many computers support this algorithm in hardware, making it faster. The algorithm 32 (8-bit) characters are output, then  $2^{256} \approx 10^{77}$  or possible digest values. SHA-256 and other algorithms are specified in Federal Information Processing Standards (FIPS) 180-4 [4]. The NIST protection hash site includes FIPS interpretation of all NIST-approved hash algorithms.

Input Text	SHA-256 Digest Value
1	0x6b86b273ff34fcee19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
2	0xd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
Hello, World!	0xdffd6021bb2bd5b0af676290809ec3a53191dd81c7f70a4b28688a362182986f

Table 1.1 - Examples of Inputs and SHA-256 Digest Values

All data in the system is protected. The chain of the block is securely encrypted, which opens the way for obtaining reliable and open information. That is, through the information in the block you can see all the "millionaires". But it's impossible to know who owns it. A special key is used for confirmation. It depends on it, the user will be identified by the system or not.

Security and reliability blockchain is built on special keys, through which the process of verifying the correctness and truthfulness of information is simplified. The cryptographic key itself is a group of letters and numbers, the calculation of which is done with the use of a specially created algorithm, called the hash function [4]. In this case, the user has only one key, which has two different qualities:

Firstly, having a key on your hands, you will not be able to learn the primary (initial) information;

Secondly, it is impossible to select another data packet that allows creating the same key.

The presence of the key on the hands does not mean anything. A person who has a key can not harm the system or another user. On the other hand, the study of available information allows you to verify that the data corresponds to a particular key. In addition, even with a small adjustment of the data, the key will also be changed [3].

On the foundation of the technologies under consideration, systems could be started for carrying out different operations, whether or not the participants in the transaction aren't familiar and also have no reason at all or maybe preconditions for mutual loyalty. Because the amount of potential input values and potential outputs

is actually a finite amount of excitable values,  $\text{hash}(x) = \text{hash}(y)$  (ie, the hash of {two|2} different inputs is actually digested simultaneously). Nonetheless, for virtually any such input Y and X, it's possible to develop the very same pipeline as the blockchain process (in this particular instance, both areas have legitimate blocking transactions), and additionally, each and every location also needs to be there. It's believed that the time of making use of another hash algorithm (SHA-256) is actually a collision resistance because the algorithm were implemented to discover the collision between SHA-256, on average  $2^{128}$  occasions.

Blockchain Technologies takes a listing of creat and transactions "fingerprints" (digest fingerprints). Any individual with exactly the same transaction list is able to produce an accurate fingerprint. In case you modify the worth of an accounting in the listing, decrypt it to be a block change, which makes it much easier to find changes that are minor.

1.1.2 Transactions. Record the transfer of assets involving a transaction gathering (digital currency, yarn, etc.) Each point in time a deposit or perhaps withdrawal is actually made, it's captured within the simulated examining bank account. Table two shows an instance associated with a fictitious transaction. Each block within the blockchain has numerous transactions. The transaction requires no less than one of the following info fields, but could be more:

- amount (total quantity of digital assets utilized for transmission) - feedback. The list of assets is actually transferred to the selection (which equates to the amount of the entire value) Note that every single digital resource is uniquely identified as well as the various other advantage can be from values which are different. Nonetheless, it can't add or perhaps delete existing assets from alternative digital assets, digital assets of several brand new assets (low values can certainly be split in each), or even might be mixed to generate brand new digital assets [12].

- outputs. Each output worth of an account which receives digital assets is actually transferred to the brand new proprietor, the identity of the brand new owner, along with a set situations to meet the brand new owner. If the offered digital assets exceed the needed extra financial resources refunded to the sender (this is actually the "change" mechanism)

- transaction ID/Hash. Uses a block to take custom identifiers for each transaction ID of a specific transaction, like the hash of a particular block.

	Input	Output	Amount	Total
<b>Transaction ID: 0xa1b2c3</b>	Account A	Account B	0.0321	
		Account C	2.5000	
				2.5321

Figure 1.2 - Example Transaction

1.1.3 Asymmetric Key Cryptography. The blockchain strategy for making use of a pair of asymmetric scripts is actually an open and mathematically related standard method for asymmetric encryption-1 (public/private primary factor encryption also referred to as private key) like a private key. The public is able to

help make it public without minimizing the security of the process, although it's essential to maintain the password protection of the information.

Even if there's a connection between two keys, the private key cannot be established based on the understanding of the public key. Asymmetric key encryption uses totally different keys to combine certain functions based on which service. For instance, when digitally signing data, the encryption algorithm uses a private key to log in. The relevant public key could then be applied to validate the signature.

Use asymmetric key encryption of blockchain systems: - the private key element is actually used for digital transactions.

- public key to receive the address, public key pair consumed in some cases to create multiple addresses public.

- the public key is actually used to verify the signature developed utilizing the private key.

- provide the capability to verify {that|which} Asymmetric transfers value to other users that have a private key that can verify the encrypted value [9].

1.1.4 Generate Address and Address. The address of the user, which is the string seems to have the user 's public hash, several extra data (errors are actually detected). These cards are used to send as well as receive digital assets. Many blockchain systems use a single system address as the "to" as well as "from" endpoints. The address is actually shorter than the public key, and isn't a secret. In order to create an address, it's normally a public key that may convert the content to a hash: Public Key> Hash Function> Address.

The user can create multiple private/public key pairs, to ensure that different tolerant anonymities are actually allowed in the terminology of the desired language. Resolve the user 's "identity" problem of blocking the public, and generally the address is going to be converted to a QR code for ease of use. When a block allocates digital assets, they're implemented by assigning them to an address. In order to invest digital assets, users need to have a private key connected with the address. By digitally signing the transaction which has the private key element, the transaction can be validated with the public key.

Most customers that block the system don't process their very own private keys, but application, often known as wallets, is actually stored securely. The wallet is able to keep private keys, public keys, as well as associated addresses. The e wallet software also can compute the user 's total assets [14].

The private key is often completed utilizing a secure arbitrary feature, it's not possible to rebuild the M 454, that's whether the user can't be privately connected to the loss of any property loss related to the blockchain. In the situation of private chain, the attacker can't have full access to each of the assets managed by the private key.

Private key security is extremely important, most customers use specific security hardware to keep it, blockchain is actually a set of private engineering. When "Bitcoin has been stolen..." This's the discovery or perhaps news channel, meaning that the private key was discovered to use cash to transfer cash to a brand

new account, but the method wasn't destroyed. Please note that data can't be changed properly and can't be revoked as soon as the criminal's stolen private element as well as publicly linked finances have been transferred to the next account.

1.1.5 Ledger. The account holder is a set of transactions. Throughout history papers have been used to observe the exchange of services and goods. Recently, papers has saved most of the large scale centralized databases in a digital manner, as well as implemented a community leader of "reliable" third parties to represent owners (ie, owners of third party deposits).

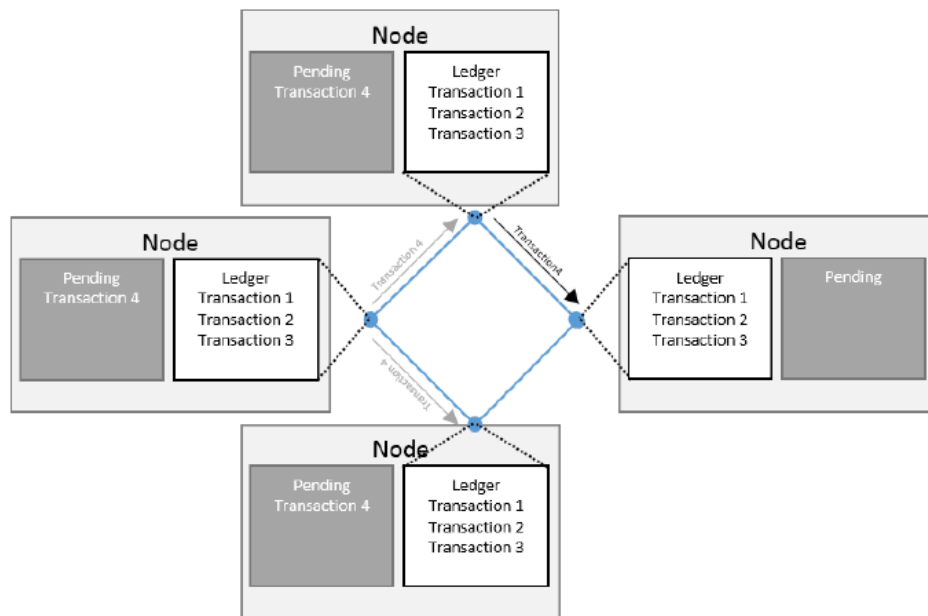


Figure 1.3 - A simple network maintaining a copy of a ledger across nodes

Central ledgers may encounter the following error:

- They may be lost or destroyed. Users should trust that the owner is backing up the system correctly.
- The transaction is invalid; the user needs to believe that the owner is validating each transaction received.
- The transaction list cannot be completed; the user thinks that the owner includes all valid transactions.
- The data may have changed; the user must believe that the owner will not change the previous transaction

That is, the backup data that accepts the transaction is in the best interest of any centralized account of the account, including all valid transactions, and the history does not change.

The ledgers used by blockchain can reduce these problems by using distributed systems. One of the reasons is that the block channel will be copied and will be distributed in each node of the system.

1.1.6 Blocks. Account holders can submit their accounts by sending users to some nodes participating in the blocking channel. The submitted activity is

promoted to other nodes in the network (but not to destroy transactions) and does not wait for distributed transactions in the queue or transaction pool unless they are connected to the mining node in blocking situations. Mining nodes are subsets of blockchain nodes that issue new blocks. When the mine site node block [2] is published, the transactions in the block are added.

There is a set of verified transactions in one block. The fund's issuer determines "validity" by ensuring that each password is signed by the transaction (as listed in the "input" value of the fund).

It verifies the use of private key funds to trade to sign the available funds. Other mining nodes will check the validity of all transactions in the published block, and if there are any illegal transactions, the block will not be accepted.

The actual construction of this block is slightly more complicated (Figure 1.4).

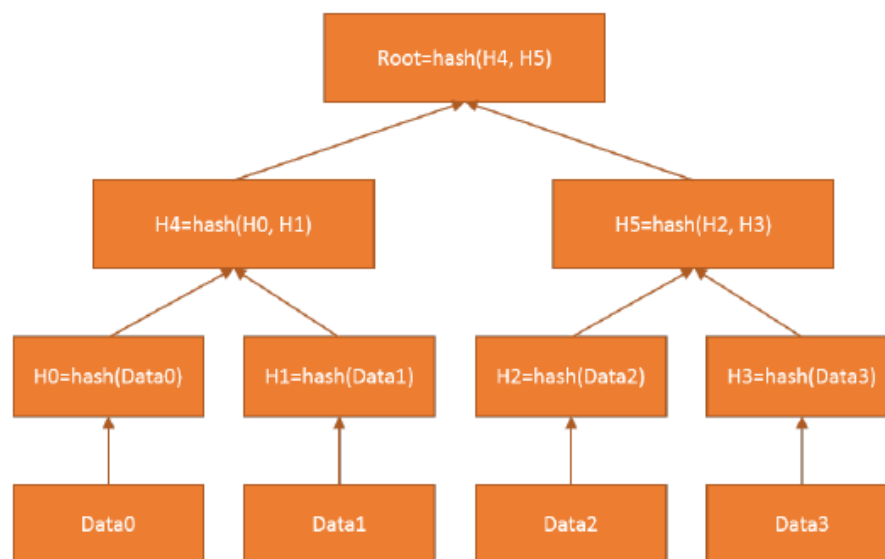


Figure 1.4 – Example of a Merkle Tree

The data field contains the following:

- current block hash
- previous block hash
- the original hash of the Merkle tree (defined below)
- timestamp
- block size
- decompressed values, which are some of the tools that mining nodes have to handle in order to solve hashing problems. This gives them the right to publish blocks.
- list of transactions included in this block

As long as there is an unusual root (Merkle tree root hash), the hash value of the Merkle tree data is combined. Root is an effective mechanism for summarizing transactions in blocks and confirming transactions in blocks. This combination ensures that the data sent in the distributed network is valid because any changes to

the underlying data will be discovered and discarded. The merkle tree in Figure 5 shows:

- The bottom line indicates data compression, which is the transaction data in the block chain

- From the second line, it shows that the data hash has been created.

- The second line collects rigorous data, and the third line collects hash data.

Finally, the top row represents the root hash, which corresponds to H4 and H5, and is. Original hash has all previous combinations and hash hashes [24].

1.1.7 Chaining blocks. By adding the first block's head, the blocks are added together in each block. If the previously released block has changed, it will have different hash values. Next, all subsequent blocks will have different hashes because it includes the hash of the previous block. This makes it easy to find and reject any changes in previously published blocks. In Figure 5, we see the general chain.

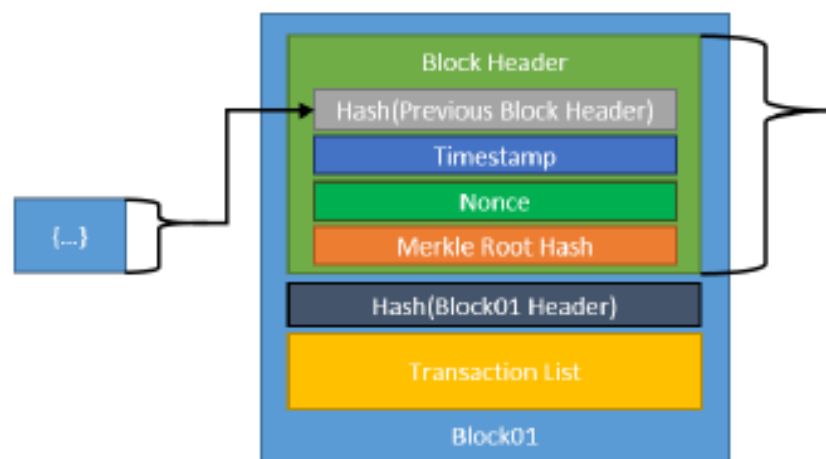


Figure 1.5 – Piece of chain of blocks

Some blockchain systems must sacrifice building the next block - such as spending time and effort, or gaining privileges. For systems that take time and effort, mining nodes compute many random values when trying to solve a problem.

## 1.2 Basic cryptocurrencies

1.2.1 Bitcoin. It is a digital cash system known as the pioneer of using blockchain. Create new blocks using SHA-256 hashes every 10 minutes to group them together. This is a proof of operating system, it does not need to include mine nodes in its block, for example, the hash of the block is less than some predefined difficulty value.

The difficulty of attempting to achieve a 10-minute block creation goal has been adjusted higher or lower. The payment of the block that is technically rewarded by the receiving of funds through the mineral node is released. This fee is designed for the small fee for each transaction, but it may be getting larger due to a sufficient backlog of pending transactions [28].

Adding more transaction costs means adding more credit to the blocking channel. Initially, the mining node gets 50 bitcoin for each block, and only a few of them are blocks. E.g. In July 2016, the block mining bonus was 12.5 Bitcoin. With every blockchain agreement, this return will be reduced by 210,000 blocks (approximately four years) and will be reduced by zero after 21 billion bits [13].

The bitcoin mine will continue to be in these places, but the bonus will be earned through full transaction fees. The last interesting technical note is that every transaction in blockchain has a piece of code written in a script called a code that indicates a simple procedure for specifying a transaction. It has no loops and is highly limited in terms of functionality. Bitcoin transactions today use only a small part of the scripting capabilities available to the script. In fact, with most Bitcoin transactions, only a few template codes are used to fund the activities of the parties.

In short, we talked about what Wikipedia is. But let's enter Bitcoin and find the right thing.

Bitcoin requires two basic devices to work: blocking and digging.

These transactions are in "blocks", which is the safest way to encrypt information and adds another group together.

Blockchain is opened at any time to everyone and most networks can be changed, but the desire for computing power. This means that it is difficult to emulate and do not break into human error, not a point of failure.



Figure 1.6 – Step by step process in blockchain

1. Miners merged directly into the up transaction in 'block'.
2. Block is protected in encryption and it is currently associated with present blocks.
3. Miners earned the block prize plus they could provide it back to the market.

Mining are actually the procedure needed to continue to be safe in each and every block, while brand-new cryptocurrency units are actually released. These products are actually also known as "block price." In bitcoin's case, the block reward is currently 12.5 Bitcoin, although this's done every 4 years or completed.

The role of the miner is actually solving this complicated algorithm - by switching the complexity of the algorithm, a continuous operation can be made easier or perhaps more challenging, and minerals are able to ensure that their processing time remains approximately the same [3]. Due to the important role of theirs in the network, miners have control which is important over bitcoin, specifically mining is now a huge business [11].

Once these tokens are released, they can be released through free exchange and can be stored in a digital wallet.

What's bitcoin fork?

An interruption happens when blockchain is actually split into 2 parts and then two individual data records are actually produced. Bitcoin relied on my network to keep on using 1 of them and made the decision to give up.

The fork group mining program is actually misbehaving as well as inhibits the required application updates. The 2 major types are hard forks and soft forks.

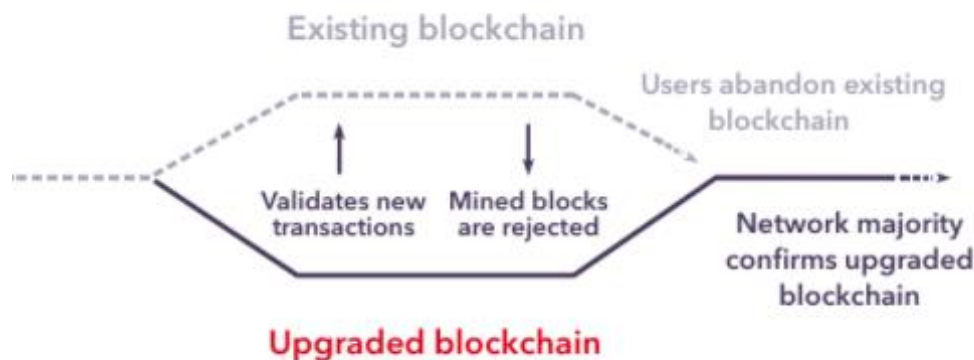


Figure 1.7 – Soft fork

Soft fork: The up-graded blockchain is currently accountable for confirming all the transactions (blocks), but current block libraries will identify and capture transactions. Be aware that this only applies to one way: The upgraded block library can't spot any of the mined blocks by using the software of the current block.

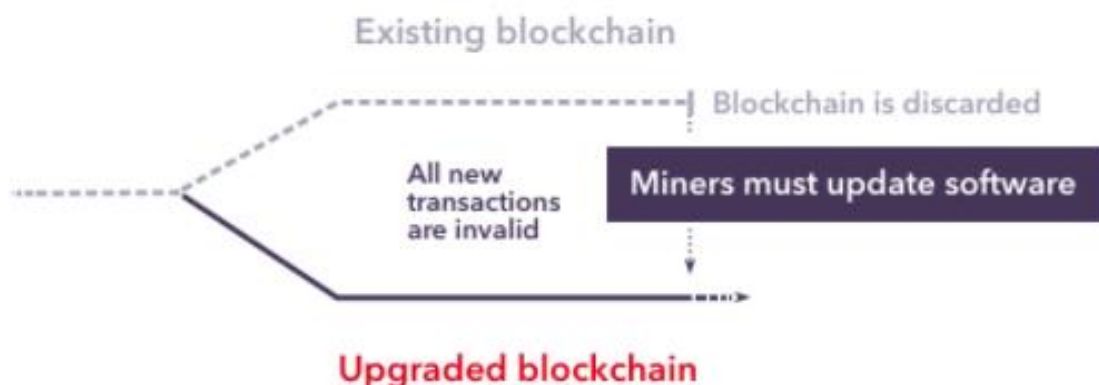


Figure 1.8 – Hard fork

Hard Forks. The upgraded blockchain is actually responsible for accepting all transactions so far, but currently existing large fragments can't recognize these blocks as legitimate or perhaps even record them. This means that all users of the old program need to be updated to use the upgraded blockchain.

In most cases, there's certainly no delay within the release of funk. Nevertheless, in recent times, the differences in the political operations or perhaps features of passwords have been ignored. Bitcoin's highest profile illustration is actually Bitcoin cash, which is actually work which is hard both in bitcoin and bitcoin cash. In the long run, there were 2 separate cryptocurrencies, Bitcoin and Wikipedia cash, though the same transaction past wasn't sold until July 2017.

1.2.2 Ethereum. What's ethereum (ethereum) - few know, despite the point that this particular crypto currency is pretty well known in the virtual environment. The ethereum project is actually a platform for creating a decentralized online service that works on the basis of blockbustor and smart contracts. A good example of the project of the ethereum system and new crypto currencies was presented by Vitalik Buterin at the tail end of 2013. But the principle is not nevertheless a ready project, as well as it was launched exclusively on July 30, 2015.

By the way, surprisingly, Buterin is actually the founding father of Bitcoin Magazine, because the interest of his in the development of his crypto currency is actually understandable. Being a result, the thought was came to the realization as a single, decentralized virtual device. The truth is, ethereum (eth) is actually a simpler and sophisticated more variant of the blockbustor. And this crypto currency can be lucrative to extract, in addition to bitcoin, specifically since it's much cheaper, and the extraction is actually simpler. As for buying crypto currency, it can be done on pretty much the most prominent world exchanges, that also speaks inside favor of this prospect of this coin [18]. Of course, crypto-currencies after bitcoin appear quite often, but they do not differ either in stability or in prospects. When the crypto currency ethereum appeared, many people paid attention to it, as it has profitable differences from the technology of the blockade - the so-called smart contract, which will significantly protect the transactions and at the same time simplify them [19].

Of course, many users are interested in the maximum number of coins ethereum. Their number is also limited, as in the bitcoin chain, but in this case the maximum number of coins is 1.800.000.000 ETH.

Ethereum is an open platform, because it is interested in more people and even large organizations. For example, such large software developers as Microsoft, IBM and Acronis are interested in the system. In addition, the Russian crypto-ethereum and a platform for it interested in such companies.

There is also an international charitable organization UNICEF among the interested persons.

So, such a crypto currency as ethereum has a high chance of becoming, if not the second bitcoin, then it is exactly to catch up with it at a price.

By the way, figuring out what eth is, you can not ignore the history of the creation of ethereum, which was developed by Russian specialists. As already mentioned, the history of creation began in 2013, but the present formation occurred a little later. The first point of currency formation was the publication of Buterin.

Further, Ethereum described in his work Gavin Wood. In addition, during the same time this particular technique was described as the coming version of bitcoin

and was known as Bitcoin 2.0. So, clarifying exactly who the founder of this etherium, unequivocally specify the title of Vitaly Buterin [21].

In 2014, they started raising resources for the development of crowdfunding.

The economic development of Ethereum started after the first public offering. Then 31 591 bitcoins (at this time 18 439 086 dollars) ended up being exchanged for 60 102 216 ether. This attracted the focus of banks.

The Ethereum block platform was officially launched on July 30, 2015.

March 14, 2016 there was a second state of the art, as Ethereum came out of the first alpha model of Frontier. The truth would be that before this, developers could not guarantee complete network security. Obviously, the new variation of the protocol wasn't particularly stable, however guaranteed more or perhaps less secure operation.

So the date when the etherium appeared, 2015 can be considered, and in 2016 this crypto currency and system became possible to use without fears that everything will collapse at any moment. In addition, other crypto-currencies can be used primarily for financial transactions, while ether is actively used as a means for exchanging resources, etc., which allows him to gain more and more admirers.



Figure 1.9 – Ethereum Smart Contracts

Ethereum is supported by smart contracts. A smart contract is an irreversible action that will be executed as soon as there are appropriate conditions for this. If in simple words to explain what a smart contract is etherium, then a smart contract is a program that is both automatic and autonomous. It can not be intervened from outside, because it is completely safe.

This technology works as follows:

- The creator of the contract puts the air on two positions, planning a transaction.
- a smart contract is being created, it is already impossible to make changes to it;
- the system checks whether the customer can perform the planned action with coin eth;

- if the action is feasible, the system transfers the broadcast to the account that was originally specified in the contract;
- the deal is over;

Everything is quite simple, and most importantly, this procedure is absolutely transparent, which attracts users. Thus, the system of smart contracts claims the title of an ideal system of contractual relations. Its development is unlimited, because it is constantly being improved, making various transactions as accessible to users as possible.

### 1.3 Hyperledger technology

Hyperledger is a set of projects aimed at building enterprise-class, open source distributed ledgers. The Hyperledger project is supported and hosted by the Linux Foundation. Although hosted by the Linux Foundation, each project is developed and delivered from a variety of sources. The Hyperledger Project has many projects that provide a blockchain platform to solve each specific problem.

Some of Hyperledger's executive governance is the leader committee. It has more than 10 officials, most of whom have decades of open source experience and they have links to most industries. For its members, Hyperledger not only provides technical knowledge and software framework, but also provides industry and developers [15].

This decision is separate from the super skier's strategic goals, and usually comes from a rich solution developed in the currency-based block mechanism used by the industry of Blockchain technology. It may be more boring, but it can be easier with technology.

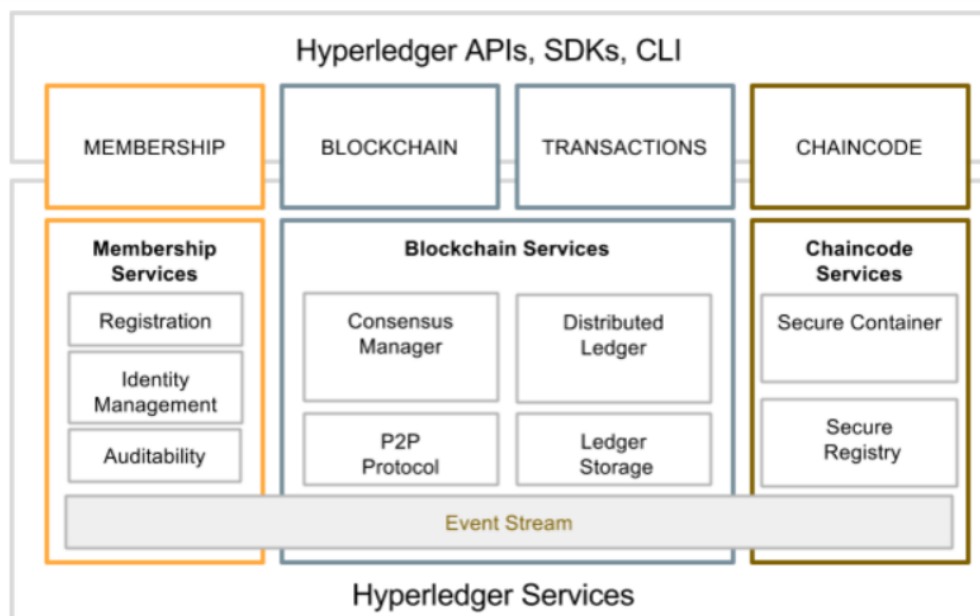


Figure 1.10 –Hyperledger reference architecture

The principal requirement of Hyperledger is actually a modular combination. Different services must be hooked up and also played, and users are going to be

able to effortlessly remove and put in modules depending on the business portfolio of yours.

Furthermore, the value of privacy and trading of smart contracts is likewise essential. Hyperledger has a multitude of encryption protocols as well as algorithms that you are able to choose and perform without sacrificing performance.

There's also a demand for a flexible PKI (Public Key Infrastructure) model which could be used to regulate access control functions. It's anticipated that the capabilities and types of cryptography will differ according to the needs of users.

The possibility of auditing is another demand. It's anticipated that all identities, related transactions, and any changes will continue to be audit trailed.

So as to make sure the relationship between the various tasks, all big females ought to have the succeeding set of conditions. It assumes that HyperPluger is not only lightweight, but additionally at the degree of code, libraries, and APIs.

Hyperledger's "Umbrella Strategy" encourages inkjet technology and turns into a part of certified blockaine technology, libraries, frameworks, applications and interfaces [10].

Currently, Hyperledger is actually the host of this following items: 1. Hyperledger Sawtooth: This's a modular blockchain suite developed by Intel, a brand new synthesis algorithm called Proof of Work (POW).

2. Hyperledger Iroha: Iroha is actually a project of a second tier company in Japan which can easily increase frameworks to the blockchain.

3. Hyperledger Fabric: IBM has this particular project. The framework is actually a plug and accessories BlackBean technological know-how which develops superior blocking applications.

4. Hyperledger Burrow: This undertaking has developed an approved smart contract machine that contains the specific details of the blockchain.

1.3.1 Hyperledger Fabric. A modular, permissible block is able to run sensible contracts (called chaincodes). Fabric blockchain up front contributed to the Hyperledger Project Department assets as well as IBM.

It's not really a blockchain framework but a modular architecture which supports the enhancement of block based solutions. BlackBerrys are able to have plug-and-play playlists including concurrency and subscription services , as well as different fiber optic components. The architecture is designed to put together a plan to integrate the activity itself right into a personal blocking network which could manage over 1000 transactions per second [11].

The framework is enabled on the go. This is done to make the Consortium Blockchain effective at different levels of licenses. Fabric chain code relies heavily on the smart contract system, which is a partner in the network running Docker Container

The structure is fewer than the common structure nodes, and the data is calculated in parallel, which makes the fabric better than the public. In addition, the infrastructure supports confidential data, so if they are found in public blocks, their members will be more confidential [27].

Clever contracts in Fabric are labeled as chaincode. The Hyperledger Fabric networking consists of "peer-to-peer nodes" which perform obstruct code, access register data, assistance transactions, as well as an application interface.

Fabric is actually created for tasks that need distributed register engineering (DLT). Supports chaincode in Go (Golang), Javascript and Java (via Hyperledger Composer, or even from edition 1.1) and consequently possibly much more adaptable compared to the language of sensible contracts.

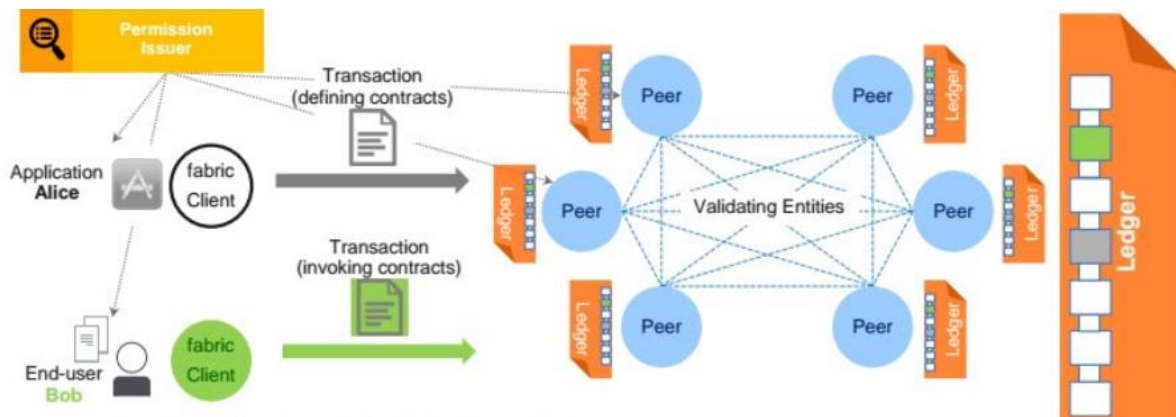


Figure 1.11 – Hyperledger Fabric model

In the lack of basic currency, this particular framework allows users to define assets with customers and make use of them with fabric combinations. The chain code of this system is actually similar to the business logic of the attribute of the smart contract framework, the so called asset state, outlined by the rules to read as well as change. The same as the thermal medium system, the rarely used output set remains unchanged, such as the continuation of bitcons, but the blocking state isn't limited to transaction data [10].

Besides the public cryptocurrencies structural block, participants are able to produce distinct channels for their assets, and separate and separate therefore transactions from account holders. In this way, the chain code necessary reading and alter the state of the asset is going to be established for the participants in a particular business case. [11] Similar to a good chat program, the structure blocking program allows the user to open and participate in 2 kinds of private interactions [28].

The fundamental principle:

- permissions system, powerful identity management
- differentiation between legitimate dealers and users
- users deploy brand new code (chain codes) and also distribute as well as start transactions;
- the certifier assesses the impact of the unit and the transaction on the new version of the account holder;
- laser = Total Transactions Hash (Global Status)

- pluggable consensus protocol; Fabric licenses as well as privacy are highly flexible, therefore the higher levels of network shareholder work can achieve high levels of liquidity.

IBM Fabric has become a big project developed by Blockcheng. The IT giant uses the structure with the assistance of a variety of projects and many business partners.

Let us demonstrate the six technical advantages of Hyperlager fabric for blockchain networks.

The Hyperledger architecture is called a web-enabled platform and all participants have a familiar identity. When considering allowed networks, you should consider whether the issues used in your blockchain must comply with data protection regulations. In most cases, especially in the financial sector and the healthcare industry, they are subject to data protection laws that belong to network members and should know who is accessing specific data.

For example, consider private equity firms. By definition, private equity is not publicly traded on stock exchanges, and its investors are usually venture capital firms, private equity firms or angel investors. Participants in this network need to be identified and have credibility in the capital so that they can participate in the blockchain [30].

Hyperledger Fabric is built on a modular architecture that separates transaction processing in three steps: distributed logic processing and protocols ("checksum"), transaction processing sequences, and transaction processing authentication and commitment.

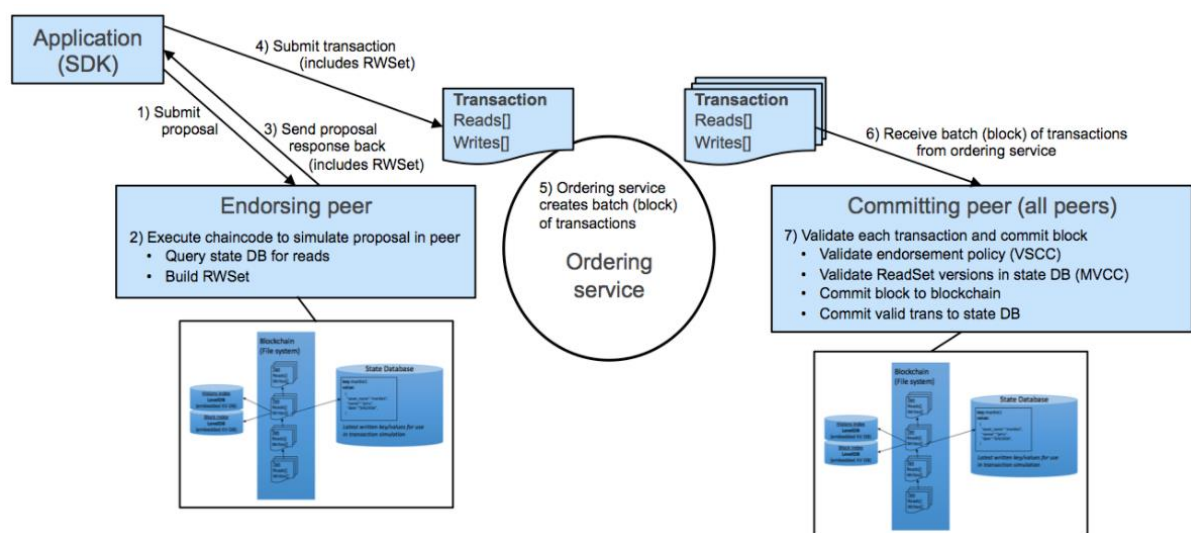


Figure 1.15 - Transaction lifecycle in v1.0 of Hyperledger Fabric

From the left aspect of this diagram: - Submit the peers of transaction proofs via the application.

- The support policy outlines how much and/or the spokesperson needs to log in the proposal. Supporters activate scans to create read/write configurations in networking peers to simulate provisioning,

- Later supporters send again recommendations to proposals which sign the application.
- The software submits signatures and transactions in the subscription system
- Deliver batch or perhaps drip blocks and send them together.
- When an investment colleague receives a batch of transactions, for each transaction
  - It verifies that the verification/reading kit has been examined as well as checked as well as found conflicting transactions. In case each of those examinations pass, the block should be an account, as well as the status of each transaction is actually shown in the updated state database.

As a result of business competitiveness, the control of this privacy of the safety law and individual info determines the confidentiality of specific details substances which could be obtained through data partitioning on the info block channel. The channels supported by the Hyperlagleg structure merely let those parties to find out what information they have to know.

Hyperledger Fabric supports the usage of one or perhaps far more networks, each of which manages transactions, conventions, and various assets between different sets of part nodes. Key features of Hyperledger Fabric is actually register of updates and requests, search based on search phrases, query ranges as well as combination essential queries, transaction history requests which are actually read only, transactions possess the signatures of every single sanctioned partner, users verify transactions with regard to the policy of application and approval of this policy, the channel register consists of a block configuration which describes the policy, command of the access checklist, along with other related info .

Channels let you create cryptographic materials from several certification authorities.

The mechanism of consensus. Consensus is ultimately achieved when the order and results of the block transactions concur to explicit examinations of the policy requirements.

Account holders are actually express transport records for block apps. Each transaction results in a set of asset value pairs crafted from billing, updates, or perhaps deletions. The unchanged factual source in V1.0 has been introduced to the peer file structure, in that will the levels DB is actually embedded.

The modularity belonging to the Hyperledger Fabric architecture makes it possible for network designers to insert elements based on the priorities of theirs, which is an advantage [10]. One of the areas where modularity is most demanding is actually to "get your own personal identity". Some multi company networks at present have identity management and importance to be reused rather compared to rebuilt. Other architectures that can be conveniently included include encryption or consolidation, several of which have their very own encryption standards.

HSM (Hardware Security Module) assistance is actually critical for protecting and managing digital keys used for intense authentication. HyperPluger Fabric provides an improved as well as unchanging PKS for production which is important, supporting other situations and identity management that demand

increased protection. So as to handle the circumstance of identity management, HSM extra protection for sensitive data and sensitive keys.

The significance of the job of the block network is usually to manage the integrity of the information and counteract the falsification of theirs. In probably the most prominent blockboy - in Bitcoin's blockade - all of the nodes compete for the appropriate to confirm the transaction, since the winner gets a reward in the type of bitcoins [21].

In the HLF, as we've previously reported, network members have various rights and conduct various activities. Thus, a distinct service is to blame for the development of the devices.

The service effort is among the most fascinating specialized details in the HLF. Unlike some other detachments, various mechanisms for reaching consensus are actually maybe here, which provide us to develop the subsequent block.

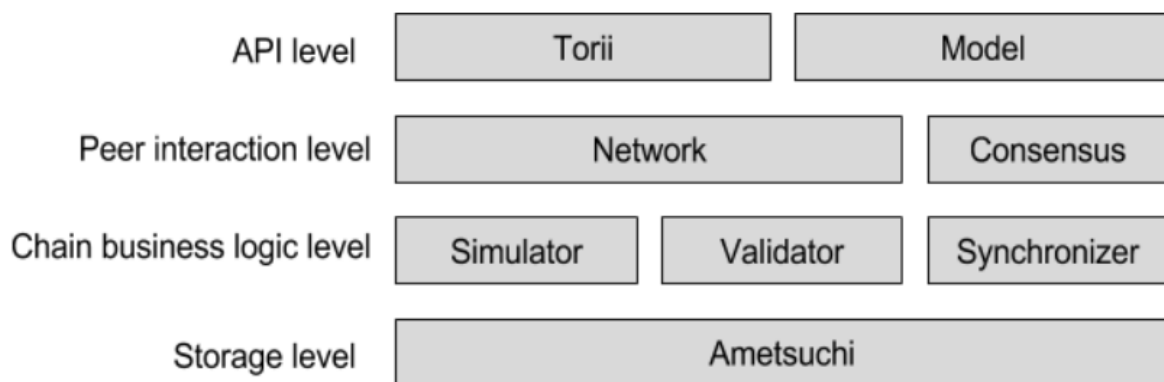


Figure 1.15 – Architectural view of Hyperledger Iroha

These parts offers input and output interfaces for Tori (Gate) clients. It's a single RPC server that clients use to communicate with friends over the network. If the client 's RPC call is actually non blocking, Torii is actually making an asynchronous server.

Collaborators reached a consensus on the contents of the web link through consensus. The consensus mechanism utilized by Iroha is actually YAC (another consensus), which is based on an actual bias based fault tolerance algorithm that votes on block hashes.

The simulator creates a short-term picture of the transaction by performing them against this particular snapshot and creating a legitimate proposal which consists of only valid transactions.

The verifier class checks the business rules as well as validity (the correct format) of this transaction or perhaps query. Hyperledger Iroha has 2 different types of authentication:

- stateless authentication is actually a fast type of authentication that determines transaction patterns as well as signatures;
- stateful validation is actually a legitimate validation, it determines the follower and also the current global state view, and that is the most recent and most

practical state in the chain, to find out in case the required business rules and policies are actually feasible; For instance, do you've money that is enough for a single account transfer?

Synchronize newbies to the synchronization structure or perhaps assistance temporarily disconnect peers [26].

The Hyperledger Iroha Network has 3 primary parts: Customers are actually: "Transactions" of "data" refer to the query statistics and they've the proper to access/permit to point out the change behavior. For instance, in a transaction, the user is able to transfer money to 3 individuals (3 independent orders). Nonetheless, in case he doesn't have sufficient money, the whole transaction is going to be rejected.

Coworkers are actually processing the current status along with their shared accounts. A peer has a single entity in the network and has trust, identity, and address. Hyperlager erosion has been developed to ensure that peers can be computers or perhaps computing clusters, which means a variety of computer systems are actually utilized for storage, authentication, indexing, and peer-to-peer logic.

In order to purchase service order transaction info There are actually alternatives for the algorithm used by the subscription service. Kafka is regarded as a good candidate. It's important to point out that in case Kafka or even any other distributed solution has been used, it's already gathered together; if not, it is going to have a point of failure.

These transactions are actually protected for cryptographic hashing and make use of a signed public/private key combination for signing and verification. Merkel Pella summarizes the transaction history summary, hence any attempt to edit and / or replace a previous transaction can be accomplished effectively and efficiently therefore that all subsequent transactions probably can be resumed. The use of blockchain remains in the early stages of its, however, it is based on considerable understanding and realistic cryptographic principles.

Looking ahead, blockchain might be a brand new application which may be used to fix new issues. The blockchain is most likely among probably the largest financial institutions in the world. They might have to change the value of theirs or perhaps completely change the behavior of theirs to be able to focus on establishing a worth assessment platform [12].

Companies that have to register public issues through specialized technologies such as land ownership, birth or marriage records should think about resolving the problems of theirs. Blockchain also offers the power to store as well as record supply chain records. With this particular feature, blockchain may end up being effective tools for developing decentralized secured applications and networks to them [17].

Blockchain is able to record every stage of product life. At the moment of delivery, they've already shipped and sent to the store which has been established and bought at the end of the client.

There might be new industries, such as digital notifications, which allow someone to access specific info by logging a hash within bitcoin. There are lots of potential uses and possibilities for blockchain technology.

Blockchain technology could disrupt numerous industries. In order to stay away from losing opportunities and avoid unexpected surprises, organizations have to check out in case blockchain is able to aid them.

The concept of blocking engineering is as easy as they can - it is an enormous public data base which works without centralized control. In the situation of bitcoin, the transaction checks are actually done by the so called miners participants of the ca, and they verify the authenticity of the committed steps, and then type blocks of transaction records.

## **2 Practical realization**

### **2.1 Purpose and goals of the project**

Within the framework of the final qualification work the following tasks should be considered:

1. Creation of a blockchain application based on the Hyperledger Sawtooth software framework capable of storing data is decentralized.
2. Demonstration of an example of this blockchain application for the international system of collection and registration of samples of athletes in the framework of independent testing (doping control) - ADAMS.
3. Registration and authentication of the basic data used to register the athlete's sample, such as: athlete's name, doping officer's name, sample number, mission code, substance density)
4. Create a multi-user data transfer mode using the application block based on the Hyperledger Sawtooth platform.
5. Description of the architecture of the system: the components used for data logging, the assignment of components of the application block, the registration of users.
6. Calculation of the required resources to create and provide technical support for this block of applications.
7. Evaluation of the reliability and availability of this system by several parameters.

For this project, it is necessary to provide for the user to act solely on his behalf. Anonymity in this case is unacceptable.

The main tasks of creating blocking applications on the basis of the Hyperledger Sawtooth software framework are:

1. Transfer of authentic data used for the registration of an athlete and a sample in the international system ADAMS with the inability to implement subsequent changes in the registry.

2. Minimization of errors in the process of filling the form of doping control.
3. Database storage is decentralized with the ability to browse transaction histories.

## **2.2 Justification of the Hyperledger Sawtooth**

Based on the project of mine, we claimed to build up blockchain program for World Anti Doping Agency (WADA). For the objective we've selected Hyperledger Sawtooth based mostly on open source operating system Ubuntu.

Hyperledger Sawtooth, contributed by Intel, is actually a blockchain framework which uses a modular wedge for constructing, deploying, and working sent out ledgers. Distributed ledger treatments designed with Hyperledger Sawtooth is able to make use of a variety of opinion algorithms grounded on the dimensions of the community. It provides the Proof of Elapsed Time (PoET) opinion algorithm, which supplies the scalability of the Bitcoin blockchain without having the big power usage. PoET provides for an extremely scalable community of validator nodes. Hyperledger Sawtooth is actually created for adaptability, with assistance for each permissioned and permissionless deployments.

Sawtooth is additionally extremely modular. This specific modularity allows consortia and enterprises to make policy choices that they're best prepared to create. Sawtooth's center design enables programs to select the transaction guidelines, permissioning, and opinion algorithms which support the unique business of theirs is looking for.

First-come-first-serve foundation. Inside PoET, each and every validator is actually provided an arbitrary wait period.

"The validator with the least wait period for a specific transaction block is actually elected the leader." - [sawtooth.hyperledger.org](http://sawtooth.hyperledger.org)

Hyperledger Sawtooth is actually perfect for this particular situation due to the ability of its to observe an asset 's (in this particular case tuna) provenance as well as adventure. The capacity to batch transactions together provides for all tuna within a catch to be entered as an entire. The distributed status agreement, novel consensus algorithm, as well as decoupled company logic from the consensus covering make it possible for Miriam to be sure that she's buying tuna which has been legally caught.

Hyperledger Sawtooth components:

1. Transaction validators validate transactions.
2. Transaction families consist of "a team of activities or perhaps transaction types" (Dan Middleton) which are actually permitted on the shared ledger. Transaction family members be made up of both transaction processors (the server side logic) as well as clients (for using from Mobile applications) or web.
3. The transaction processor delivers the server side business logic that runs on assets within a network.
4. Transaction batches are actually clusters of transactions which are actually both all committed to express, or perhaps are not devoted to state.

5. The network layer is actually responsible for speaking between validators in a Hyperledger Sawtooth network, which includes performing first connectivity, peer discovery, and message handling [10].

The global express has the present state of a chain and the ledger of transaction invocations. The state for all transaction households is actually represented on each validator. The state is actually split into namespaces, which provide freedom for transaction family authors to explain, share, and reuse worldwide condition info between transaction processors [4].

Hyperledger Sawtooth transaction batches are actually clusters of transactions which are actually both all dedicated to state together, or perhaps none of the transactions are actually committed at all. Being a result, transaction batches are frequently described as an atomic device of change, since a staff of transactions are actually managed as one, and are actually dedicated to the state as one. Each and every transaction in Hyperledger Sawtooth is actually submitted within a batch. Batches are able to feature only a small amount as a single transaction.

When a transaction is generated by a client, the batch is actually submitted to the validator (which we are going to cover more in depth in the following section). Transactions are actually organized into a batch in the order they're meant to be committed. The validator then, in turn, applies each transaction within the batch, bringing about a shift in the worldwide express. The batch is actually dedicated to the state. If one transaction within the batch is actually invalid, then not one of the transactions within that batch are actually committed.

To sum things up, transaction batching allows a team of transactions to be utilized in a particular order, of course, if any are actually invalid, then not one of the transactions are actually applied. This's a strong tool which could be used by many enterprise solutions, as it allows control and efficiency greater for end users.

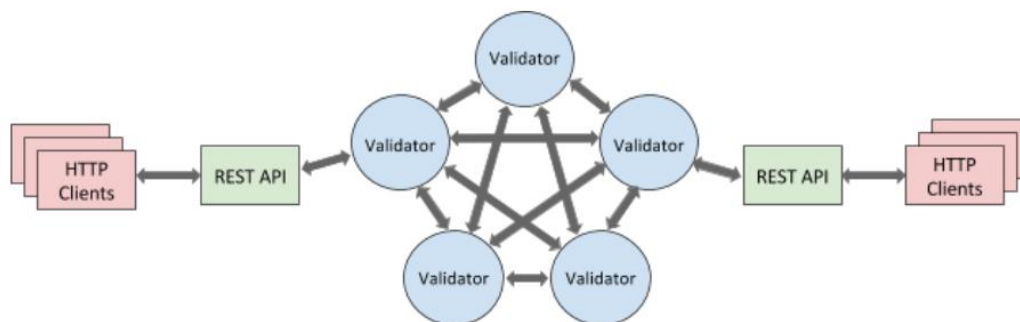


Figure 2.1 – Hyperlegder Sawtooth structure

In any blockchain network, modifying the global state requires creating and applying a transaction. In Hyperledger Sawtooth, validators apply blocks that cause a change in the state. More specifically, validators validate transaction blocks, and ensure that transactions result in state changes that are consistent across all participants in the network.

To start, a user creates a transaction batch and submits it to a validator via a client and REST API. The validator then checks the transaction batch and applies it if it is considered valid, resulting in a change to the state. In terms of our demonstrated scenario, Sarah, the fisherman, creates a transaction batch to record information about a group of tuna catches [11]. The validator would then apply the transactions, and the state would be updated if all appropriate attributes are present: a unique ID number, location and time of the catch, weight, and who caught the fish. If any of these elements are missing, the transactions within the batch would not be applied, and the state would not be updated.

Hyperledger is a modular platform for the Sawtooth platform to build, assign and continue the distributed registries. Distributed registrars provide digital records (for example, asset ownership) without care, central management or implementation. Using Sototh platform distributed registry technology helps in finding a series of processes and ensuring credibility. Therefore, the loet sensors can be added to any object assigned to a person, the ability to track the ownership of the object, ownership and telemetry parameters such as its location, temperature, humidity, traffic etc. The last customer can access the full data registry and rely on their accuracy and completeness.

Inside Hyperledger Sawtooth, the log maintains as well as extends the blockchain for the validator. It's in charge of validating candidate blocks, evaluating legitimate blocks to figure out in case they're the appropriate chain head, and producing brand new blocks to lengthen the chain. Transaction batches arrive at the journal, exactly where they're evaluated, validated, and put into the blockchain. Furthermore, the log resolves forks, which occur because of to disagreements over who commits a block. When blocks are actually finished, they're delivered to the ChainController for validation as well as fork resolution. To find out exactly how the components of the journal interact with one another, check out the diagram on the following page [11].

### **2.3 Structure of the framework**

Consensus in Hyperledger Sawtooth is modular, meaning that the consensus algorithm can be easily modified. Hyperledger Sawtooth provides an abstract interface that supports both PBFT and Nakamoto-style algorithms. To implement a new consensus algorithm in Hyperledger Sawtooth, you must implement the distinct interface for: block publisher, block verifier, and fork resolution.

- block publisher: Creates new candidate blocks to extend the chain.
- block verifier: Verifies that candidate blocks are published in accordance with consensus rules.
- fork resolver: Chooses which fork to use as the chain head for consensus algorithms that result in a fork.

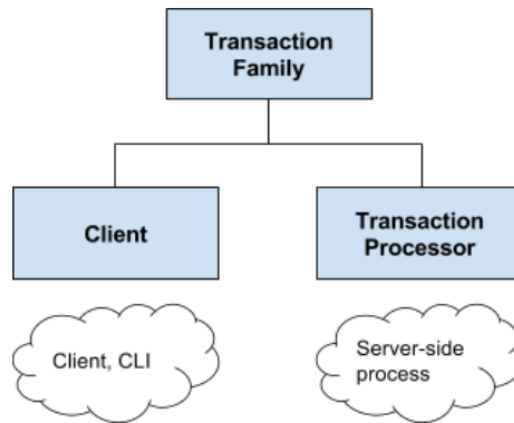


Figure 2.2 – Sawtooth transaction family

In Hyperledger Sawtooth, the data model that captures the state and the transaction language that changes the state are implemented using transaction families.

A transaction family consists of a group of operations or transaction types that are allowed on the shared ledgers. This allows for flexibility in the level of versatility and risk that exists on a network. Transaction families are often called 'safer' smart contracts, because they specify a predefined set of acceptable smart contract templates, as opposed to programming smart contracts from scratch.

A transaction processor provides the server-side business logic that operates on assets within a network. Hyperledger Sawtooth supports pluggable transaction processors, that are customizable based on the specific application. Businesses are able to develop transaction processors that do exactly what their applications need. Additionally, transaction processors can be written in a variety of languages (Java, Python, C, C++, JavaScript, and Go), allowing for ease of use and simplicity when handling assets.

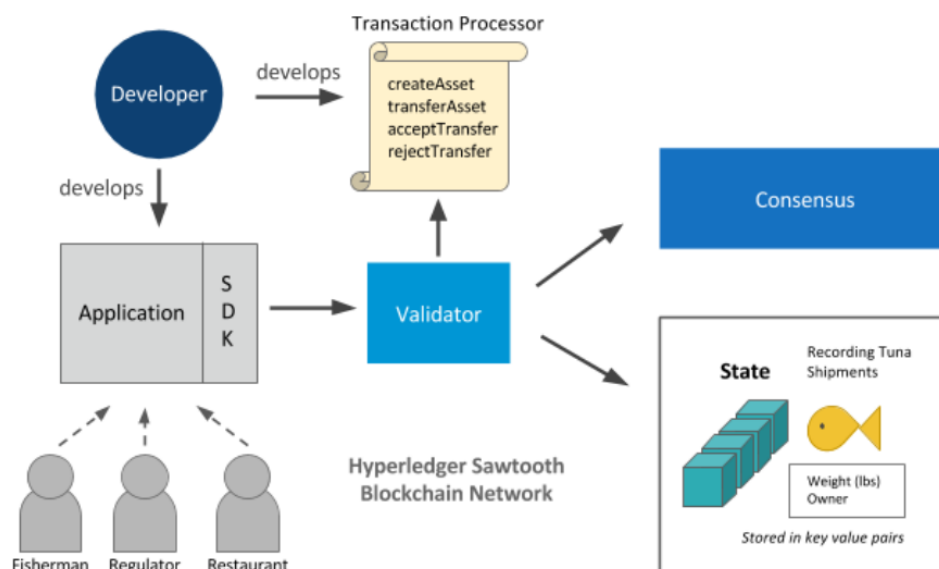


Figure 2.3 – Application flow

Hyperledger Sawtooth allows entities to securely update and read the distributed ledger without involving a central authority. Developers create application and transaction processor business logic (smart contract).

Through the client application, users (fisherman, regulator, restaurant) are able to modify the state by creating and applying transactions. Through a REST API, the client application creates a batch containing a single transaction, and submits it to the validator. The validator applies the transaction using the transaction processor, which makes a change to the state (e.g., creating a record of a tuna catch).

In Hyperledger Sawtooth, batches are clusters of transactions that are committed to state together. If one transaction in the batch cannot be committed, none of the transactions are committed. As a result, transaction batches are often described as an atomic unit of change, since a group of transactions are treated as one, and are committed to the state as one.

Every single transaction in Hyperledger Sawtooth is submitted within a batch. Batches can contain as little as a single transaction.

When a transaction is created by a client, the batch is submitted to the validator (which we will cover more in-depth in the next section). Transactions are organized into a batch in the order they are intended to be committed.

The validator then, in turn, applies each transaction within the batch, leading to a change in the global state. The batch is committed to the state. If one transaction within the batch is invalid, then none of the transactions within that batch are committed.

In summary, transaction batching allows a group of transactions to be applied in a specific order, and if any are invalid, then none of the transactions are applied. This is a powerful tool that can be utilized by many enterprise solutions, as it provides greater efficiency and control for end users.

## **2.4 Network architecture**

Sawtooth is actually a modular platform provided by Intel in April 2016, with a few main innovative developments. The concentration is actually on adaptable use in a variety of business parts, with the launch of other consensus and transaction families.

Transactions are actually separated from the consensus amount, utilizing for that purpose a brand new idea known as the transaction households. Rather than transactions which are separately connected with the ladder, transaction households are used, which offers increased flexibility as well as limitless business logic layout. Transactions stick to the patterns as well as structures identified in the transaction households.

Intel even unveiled a brand new PoET consensus algorithm, evidence of the past. The protocol arbitrarily chooses the winner, though there's no monetary incentive, as in the situation of mining.

Distributed registers present a digital history (for instance, asset ownership) which is maintained without a main authority or perhaps implementation.

Hyperledger is actually a modular platform for producing as well as driving networking laser apps. The development of software apps is actually facilitated by separating the primary process from the amount in program amount. Application developers have to know the look of the basic core process, work with the own programming language of theirs of choice, and inform the application business logic of theirs [15].

Sawtooth validators validate transactions. Validators are actually accountable for merging batches of transactions to blocks, distributing them to the ledger, and approving legitimate blocks based on the network 's opinion algorithm.

Sawtooth programs are actually distributed uses like smart protocols, that are separated from the primary framework. The transaction software describes the behavior as well as actions of the substance which offers the means for the actions put on to the household. In functional applications, the two processors (server side logic) and one or even more customers (network, CLI command type, and use of mobile programs) are actually provided.

The transaction processor offers server side internet business reason. The majority of the apps from several transaction processor nodes run a certain use situation or maybe a pair of the same networks sovotha nodane every transaction.

Batches are actually clustered together. In batches focused on the state, they are able to be a single transaction or even a number of associated transactions together. In case one transaction fails, additional transactions in this batch likewise fail.

The networking level responsible for talking between validators in a Hyperledger Sawtooth networking, which includes performing first connectivity, peer find, and note control.

Worldwide express has the present state of a chain and the ledger of transaction invocations. The condition of all apps operating on the network is shown on each node. Each transaction authentication system guarantees that transactions lead to the exact same status transitions, and it is akin to data efficient lasers for those participants in the community [18]. The state of a specific program has been split into application specific namespaces, which describes the application writer, shares the condition of the worldwide transaction processor among information, as well as enables freedom to be worn once again.

Evidence of Elapsed Time (POET) has been unanimously agreed in Hyperledger Sawtooth this brings down the application of time based algorithms for big distributed networks with algorithms as Proof of Work.

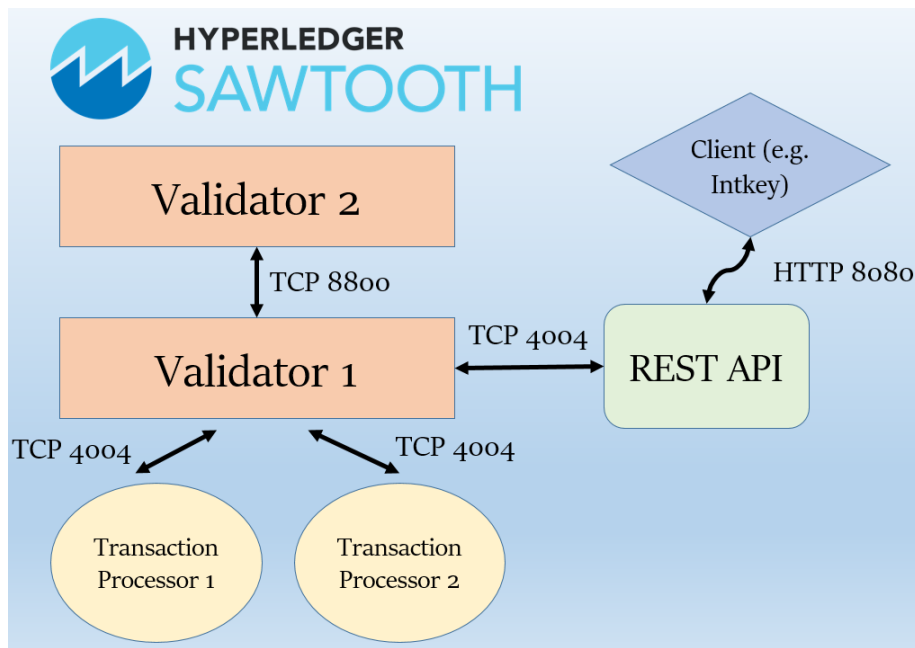


Figure 2.4 – Hyperledger Sawtooth structure

Validators as it was mentioned above are adding combinations into blocks and verifies them with the ledger. In addition, as this consensus works add them into blockchain according to it. In this case structure with only validator 1 is shown.

Transaction processors stand for business logic from the server part.

The REST API defines a set of functions to which developers can make requests and receive replies. The interaction takes place via the HTTP protocol. The advantage of this approach is the wide distribution of the HTTP protocol, so the REST API can be used from almost any programming language.

With the IntegerKey family of traditional users, users can add, increase, and decrease the value of archive entries in the state dictionary.

Transaction requests in the IntegerKey family are determined by the following values:

- descriptive trading values;
- the name of the record or the name to change;
- record settings or changed values;

The 'set' verb is used to create new records. The initial value of the record is set to the value specified in the transaction request. In the status dictionary, "inc" and "dec" are used to change the value of the existing record.

Since we have already considered the components of the main architecture of the Hyperledger Sawtooth, it is necessary to proceed to the methods of connecting these components.

Validators are connected between each other via TCP 8800 protocol also they interconnect with other components of Hyperledger Sawtooth via TCP protocol via 4004 port.

TCP is a higher-level protocol that allows applications running on various host computers to share data streams. TCP divides the data streams into chains, which are called TCP segments, and transfers them via IP. In most cases, each TCP

segment is sent in the same IP datagram. However, if necessary, TCP will split the segments into several IP datagrams that fit into the physical data frames that are used to transfer information between computers on the network. Because IP does not guarantee that datagrams will be received in the same sequence in which they were sent, TCP reassembles the TCP segments at the other end of the route to form a continuous stream of data.

HTTP (HyperText Transfer Protocol) is a seventh-layer OSI protocol for data transmission, which is based on the client-server architecture. Initially, the HTTP protocol was developed to transfer HTML documents between the server and the client using HTTP messages. Because the protocol is based on client-server interaction, it is assumed that there is a client that does HTTP requests and there is an HTTP server that processes these requests and gives the client HTTP responses. All server responses contain status codes, and all client requests have HTTP methods. This series of publications will help us understand how the client and server interact using the HTTP protocol.

In reality, there's a range of configurations and architectures of caches as well as proxy servers presently being designed or even deployed on the World wide Web; these methods include national hierarchies of proxy caches which protect the bandwidth of inter ocean channels, devices which distribute cache contents, businesses that distribute subsets of cached information on a CD ROM, etc. HTTP methods are utilized in company intranets with high speed correspondence links, and also for access with the PDA with unstable communications and low-power lines. The intent behind HTTP / 1.1 is actually maintaining a range of configurations already made with the launch of earlier types of the process.

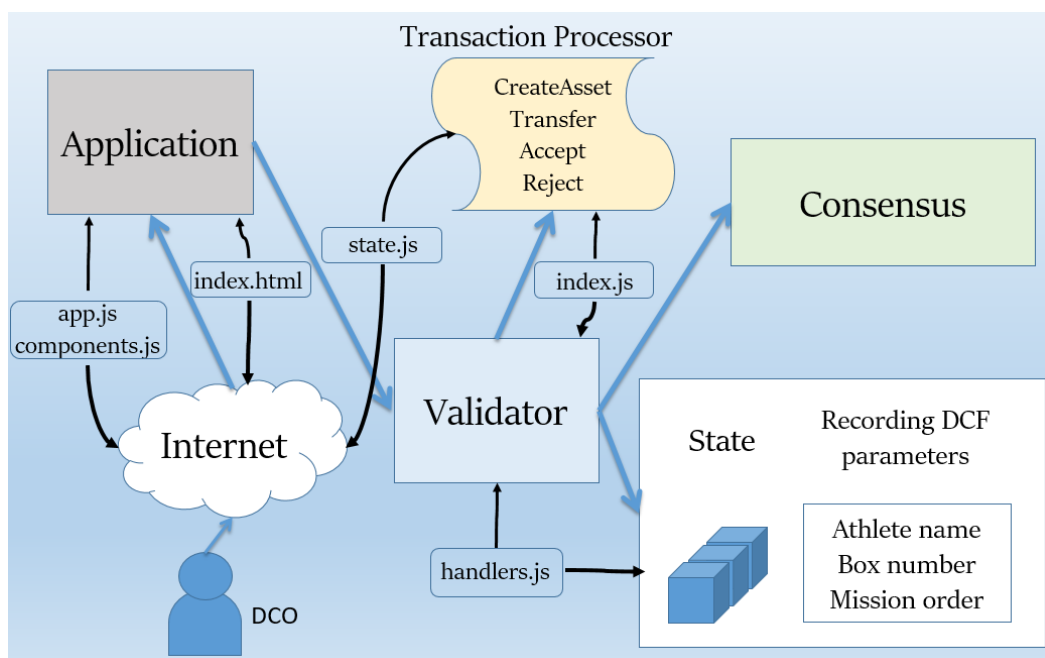


Figure 2.5 – Hyperledger Sawtooth transaction process

Let's look deeply in Hyperledger Sawtooth structure to find out what kind of programs are interconnected with components of our framework.

In the Hyperledger Sawtooth, the account holder will control the system but the transaction history will not change. An application is usually composed of two parts.

Customer application is used for blocking transactions are usually sent through the Sawtooth API.

Provides a user interface for the application. The client can be a command line interface, a web page, a mobile application, an IoT sensor, or any other type of interface capable of sending HTTP requests.

Transaction processor apply business logic code, contact the customer for authentication, and send the transaction received by the customer to the transaction processor.

Client part contains from programs, which are used to connect out program with end user. These programs consists of app.js, components.js, state.js. This folder contains the code for the blockchain browser-based client written in Javascript. After running the docker compose file, the client can be accessed at 'localhost: 8000'.

Folder containing Javascript client code that connects transaction processor backend to the client (mainly within state.js). The follow figure shows structure of state.js file. This file creates new key-pair for the application, saves key-pairs to local storage.

For validator it fetches current Sawtooth blockchain state from validator and submit signed transaction to validator.

```
const $ = require('jquery')
const {
  signer,
  BatchEncoder,
  TransactionEncoder
} = require('sawtooth-sdk/client')

// Config variables
const KEY_NAME = 'transfer-chain.keys'
const API_URL = 'http://localhost:8080'

const FAMILY = 'transfer-chain'
const VERSION = '0.0'
const PREFIX = '19d832'

// Fetch key-pairs from localStorage
const getKeys = () => {
  const storedKeys = localStorage.getItem(KEY_NAME)
  if (!storedKeys) return []

  return storedKeys.split(';').map((pair) => {
    const separated = pair.split(',')
    return {
      public: separated[0],
      private: separated[1]
    }
  })
}

// Create new key-pair
const makeKeyPair = () => {
  const privateKey = signer.makePrivateKey()
  return {
    public: signer.getPublicKey(privateKey),
    private: privateKey
  }
}
```

Figure 2.6 – Configuring variables and creating new key-pairs

Components.js and app.js manipulates html elements for the user interface.

Particularly components.js adds selected option. After it adds a new table row with the amount of cells that we need to implement in our blockchain application. Also it adds accept/reject buttons. With their help information from blockchain can be transferred between users.

```
const $ = require('jquery')

// Add select option which may be set to selected
const addOption = (parent, value, selected = false) => {
  const selectTag = selected ? ' selected' : ''
  $(parent).append(`<option value="${value}"${selectTag}>${value}</option>`)
}

// Add a new table row with any number of cells
const addRow = (parent, ...cells) => {
  const tds = cells.map(cell => `<td>${cell}</td>`).join('')
  $(parent).append(`<tr>${tds}</tr>`)
}

// Add div with accept/reject buttons
const addAction = (parent, label, action) => {
  $(parent).append(`<div>
    <span>${label}</span>
    <input class="accept btn btn-primary" type="button" value="Accept">
    <input class="reject btn btn-caution" type="button" value="Reject">
  </div>`)
}

module.exports = {
  addOption,
  addRow,
  addAction
}
```

Figure 2.7 – Adding table rows and buttons

Program that is will be illustrated below app.js contains code for blockchain application written in Javascript. This program selects users that were added to the application then in creates assets and also it can transfer and accept them.

```

// Select User
$( '[name=keySelect'] ).on( 'change', function () {
  if ( this.value === 'new' ) {
    app.user = makeKeyPair()
    app.keys.push(app.user)
    saveKeys(app.keys)
    addOption(this, app.user.public, true)
    addOption('[name=transferSelect]', app.user.public)
  } else if (this.value === 'none') {
    app.user = null
  } else {
    app.user = app.keys.find(key => key.public === this.value)
    app.refresh()
  }
})

// Create Asset
$( '#createSubmit' ).on( 'click', function () {
  var asset = $( '#createName' ).val()
  asset = asset.concat( " ", $( '#boxNo' ).val() " ", $( '#missionOrder' ).val() " ",
$( '#timeSealed' ).val() " ");
  console.log(asset);
  const boxNo = $( '#boxNo' ).val()
  const missionOrder = $( '#missionOrder' ).val()
  const timeSealed = $( '#timeSealed' ).val()
  if (asset) app.update( 'create', asset)
})

// Transfer Asset
$( '#transferSubmit' ).on( 'click', function () {
  const asset = $( '[name=assetSelect]' ).val()
  const owner = $( '[name=transferSelect]' ).val()
  if (asset && owner) app.update( 'transfer', asset, owner)
})

// Accept Asset
$( '#transferList' ).on( 'click', '.accept', function () {
  const asset = $( this ).prev().text()
  if (asset) app.update( 'accept', asset)
})

```

Figure 2.8 – Creating, transferring and accepting assets

Sawtooth is one of nine business-blocking technologies and a distributed magazine under the patronage of the Linux Foundation. The same applies to the modeling language Hyperledger Composer with support for REST API, based on JavaScript.

Platform Hyperledger Sawtooth, over which worked more than 50 software engineers, allows you to manage the chain, used in self-executing "smart" contracts when configuring the blocking configuration and determining the rights of access to the electronic journal.

The platform supports an advanced transaction mechanism that allows processing them in parallel to speed up the creation and validation of blocks.

```

const { TransactionProcessor } = require('sawtooth-sdk/processor')
const { JSONHandler } = require('./handlers')

const VALIDATOR_URL = 'tcp://localhost:4004'

// Initialize Transaction Processor
const tp = new TransactionProcessor(VALIDATOR_URL)
tp.addHandler(new JSONHandler())
tp.start()

```

Figure 2.9 – Initialization of transaction processor

Processor folder holds the transaction processor logic. Within index.js we initialize transaction processor, handlers.js contains all handlers where we define all pertinent business logic including queries and updates to blockchain state. Package.json handlers the processor's dependency: sawtooth-sdk.

```

placeholder="Enter athlete name...">
<input id="boxNo" class="form-control" type="text" placeholder="Enter
box number...">
<input id="missionOrder" class="form-control" type="text"
placeholder="Enter mission order...">
<input id="timeSealed" class="form-control" type="text"
placeholder="Enter sealed time...">
<input id="createSubmit" type="button" value="Create" class="btn btn-
primary">
</div>

<div class="form-group">
<!-- <label>Transfer Doping Control Form</label>
<select class="form-control" name="assetSelect">
<option value="none" selected>Select Doping Control Form...</option>
</select>
<select class="form-control" name="transferSelect">
<option value="none" selected>Select ADAMS/DCO recipient...</option>
</select>
<input id="transferSubmit" type="button" value="Transfer" class="btn
btn-primary">
</div>

<div class="form-group">
<label>Accept DCF</label>
<div id="transferList"></div>
</div>
-->
<div id="data">
<label>ADAMS List</label>
<table class="table table-hover">
<tr>
<th>Athlete name</th>
<th>Box number</th>
<th>Mission order</th>
<th>Time sealed</th>
<th>Doping Control Officer</th>
</tr>
<tbody id="assetList"></tbody>
</table>
</div>
</div>
<script src="dist/bundle.js"></script>
</body>
</html>

```

Figure 2.10 – Configuring parameters for html page

HTML is the language of hypertext markup, which has become very widespread on the Internet. The HTML language defines the structure of the pages that you see in the browser. Each site on the Internet uses HTML to display information.

HTML defines the structure of the pages that you see in the browser thanks to HTML tags, the browser "reads", processes them, and then displays the tags to you on the screen, but as HTML elements, with some HTML elements you can even interact with a mouse or keyboard .

To be precise from a formal point of view, it is correct to speak not an HTML page, but an HTML document, your browser communicates with the web server via the HTTP protocol, sends HTTP requests and receives the answers of the server, in the body of which contains HTML.

Like the HTTP protocol, the HTML language was developed at CERN by Tim Berners-Lee in 1991 and was originally used by scientists to exchange scientific documents. HTML clearly defined the structure of the document and allowed to highlight some features of the text of the document, thanks to this and the fact that the syntax of the HTML language was simple, it got a huge spread not only in the scientific environment, but also went to the masses.

Index.html with user interface for application. Open this file in browser to begin interfacing with Hyperledger Sawtooth. After that parameters can be easily written in blockchain application based on Hypeledger Sawtooth framework.

## 2.5 Installation process

Sawtooth SDKs offer transaction loved ones developer assistance for Javascript, Go., Java, C, C, and Python In order to produce the simple application of ours, we will make use of the Javascript SDK, which offers client performance and supplies the procedure of making alterations to the blockchain. In case you would want delving much more into these quite heavy information, complex details, stay tuned for a future program on Sawtooth.

Building an application requires a few important steps:

First, defining assets that will reside in the distributed ledger, as well as transactions that will act on these assets to change their state. Second, designing transaction logic that operates on these assets.

Now, as transactions are received by a Sawtooth node, they are forwarded to other nodes. A node is selected by a consensus model to publish a block. The block will contain any transactions that have been received and executed successfully. The block is then broadcasted to the publishing nodes. Each sawtooth node receives a published block and verifies that that block is valid. It then notifies our application of any state changes.

Hyperledger Sawtooth is a suite that permits the creation and utilization of a distributed ledger. Installing Hyperledger Sawtooth will involve adding signing keys for the software creator to our environment, including the repository that contains the code to our system, and performing a typical update/install. A Hyperledger Sawtooth validator node can be run either from pre-built Docker images, or natively using Ubuntu 16.04.

First of all we need to download Docker compose file as sawtooth-default.yaml. and transfer it to text file.

```
version: "2.1"
services:
  settings-tp:
    image: hyperledger/sawtooth-tp_settings:0.8
    container_name: sawtooth-settings-tp-default
    expose:
      - 4004
    depends_on:
      - validator
    entrypoint: settings-tp -vv tcp://validator:4004

  intkey-tp-python:
    image: hyperledger/sawtooth-tp_intkey_python:0.8
    container_name: sawtooth-intkey-tp-python-default
    expose:
      - 4004
    depends_on:
      - validator
    entrypoint: intkey-tp-python -vv tcp://validator:4004

  xo-tp-python:
    image: hyperledger/sawtooth-tp_xo_python:0.8
    container_name: sawtooth-xo-tp-python-default
    expose:
      - 4004
    depends_on:
      - validator
    entrypoint: xo-tp-python -vv tcp://validator:4004

  validator:
    image: hyperledger/sawtooth-validator:0.8
    container_name: sawtooth-validator-default
    expose:
      - 4004
    ports:
      - "4004:4004"
    # start the validator with an empty genesis batch
    entrypoint: "bash -c \"\
      sawtooth admin keygen && \
      sawtooth keygen my_key && \
      sawtooth config genesis -k /root/.sawtooth/keys/my_key.priv && \
      sawtooth admin genesis config-genesis.batch && \
      sawtooth-validator -vv \
      --endpoint tcp://validator:8800 \
      --bind component:tcp://eth0:4004 \
      --bind network:tcp://eth0:8800 \"\
    "
```

Figure 2.11 – Docker-compose.yaml file

After this, we need to start docker-compose file on Ubuntu operating system by running following command.

In a Sawtooth application, the ledger will store the state of the system, in addition to the immutable record of transactions that created that state. An application typically consists of two parts:

Client Application sends transactions to the blockchain, typically through the Sawtooth REST API. Provides a user interface for the application. A client can be a command-line interface, a web page, a mobile app, an IoT sensor, or most any other kind of interface capable of sending HTTP requests.

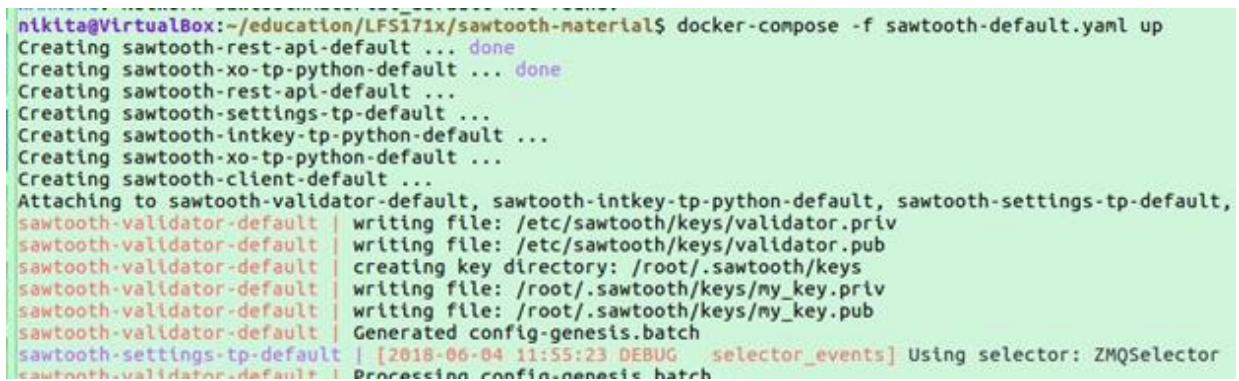
Transaction Processor encodes the business logic of the application. Communicates with the validator, which sends transactions received from the client to the transaction processor for validation.

Before running this command we need to move to the working directory.

```
cd education/LFS171x/sawtooth-material
```

After transferring we need to run the following command to start docker compose file.

```
$ docker-compose -f sawtooth-default.yaml up
```



```
nikita@VirtualBox:~/education/LFS171x/sawtooth-material$ docker-compose -f sawtooth-default.yaml up
Creating sawtooth-rest-api-default ... done
Creating sawtooth-xo-tp-python-default ... done
Creating sawtooth-rest-api-default ...
Creating sawtooth-settings-tp-default ...
Creating sawtooth-intkey-tp-python-default ...
Creating sawtooth-xo-tp-python-default ...
Creating sawtooth-client-default ...
Attaching to sawtooth-validator-default, sawtooth-intkey-tp-python-default, sawtooth-settings-tp-default,
sawtooth-validator-default | writing file: /etc/sawtooth/keys/validator.priv
sawtooth-validator-default | writing file: /etc/sawtooth/keys/validator.pub
sawtooth-validator-default | creating key directory: /root/.sawtooth/keys
sawtooth-validator-default | writing file: /root/.sawtooth/keys/my_key.priv
sawtooth-validator-default | writing file: /root/.sawtooth/keys/my_key.pub
sawtooth-validator-default | Generated config-genesis.batch
sawtooth-settings-tp-default | [2018-06-04 11:55:23 DEBUG selector_events] Using selector: ZMQSelector
sawtooth-validator-default | Processing config-genesis batch
```

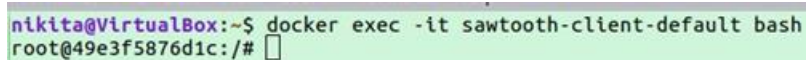
Figure 2.12 – Running docker-compose file

After running the command, we need to log into running container by running the following command:

```
$ docker exec -it sawtooth-shell-default bash
```

We will see following root folder:

```
root@75b380886502:/#
```



```
nikita@VirtualBox:~$ docker exec -it sawtooth-client-default bash
root@49e3f5876d1c:/#
```

Figure 2.13 – Running root

Our environment is now set up and you are ready to start experimenting with the network. But firstly we need to check that validator is up and running properly and reachable from the client container. For this purpose we need to run the following command:

```
$ curl http://rest-api:8008/blocks
```

```

nikita@VirtualBox:~$ docker exec -it sawtooth-client-default bash
root@49e3f5876dic:/# curl http://rest-api:8080/blocks
{
  "data": [
    {
      "batches": [
        {
          "header": {
            "signer_pubkey": "02ebce312bde38c759781b5d1f1599875fdb9af3ad8f109a87aa6263e092ebb3a7",
            "transaction_ids": [
              "5ace55671b898d03715dd85e93112cc2cb2a9a96ba8345df2a8197809e96e9935fd7a38dc06c08beef7ca1b8c7a1ac4b332257965b315911d4f444d2f5e0af50"
            ]
          },
          "header_signature": "a6c1f713230f9acde4daa525736165ab7020aaf38623fc1e1c21fd094a6960180cea53be89228729781bc583ae660ff8d877f8aa1b3ffe140eb5c0f027014283",
          "trace": false,
          "transactions": [
            {
              "header": {
                "batcher_pubkey": "02ebce312bde38c759781b5d1f1599875fdb9af3ad8f109a87aa6263e092ebb3a7",
                "dependencies": [],
                "family_name": "sawtooth_settings",
                "family_version": "1.0",
                "inputs": [
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c1c0cbf0fbcaf64c0b",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c12840f169a04216b7",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c1918142591ba4e8a7",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c12840f169a04216b7"
                ],
                "nonce": "",
                "outputs": [
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c1c0cbf0fbcaf64c0b",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c12840f169a04216b7"
                ],
                "payload_encoding": "application/protobuf",
                "payload_sha512": "497fae61053b64811df34264c7b942e88efbf7931710ff4e9e53b519694e82c618aa30c26c1a55f2f6a319d951603c69c87a885e9d678e1c4224185532aa0ca4",
                "signer_pubkey": "02ebce312bde38c759781b5d1f1599875fdb9af3ad8f109a87aa6263e092ebb3a7"
              }
            ]
          }
        ]
      ]
    }
  ]
}

```

Figure 2.14 – rest-api blocks

As you can see from the screen, there are json response objects with headers, header signatures, nonce, outputs and etc.

To run the program properly we need to check our host computer with Docker container. To obtain this doings we need to run following command in terminal windows. There is no reason whether to run it from the same directory as we did before or not. We will run it from new terminal.

\$ curl http://localhost:8008/blocks

```

nikita@VirtualBox:~$ curl http://localhost:8008/blocks
{
  "data": [
    {
      "batches": [
        {
          "header": {
            "signer_pubkey": "02ebce312bde38c759781b5d1f1599875fdb9af3ad8f109a87aa6263e092ebb3a7",
            "transaction_ids": [
              "5ace55671b898d03715dd85e93112cc2cb2a9a96ba8345df2a8197809e96e9935fd7a38dc06c08beef7ca1b8c7a1ac4b332257965b315911d4f444d2f5e0af50"
            ]
          },
          "header_signature": "a6c1f713230f9acde4daa525736165ab7020aaf38623fc1e1c21fd094a6960180cea53be89228729781bc583ae660ff8d877f8aa1b3ffe140eb5c0f027014283",
          "trace": false,
          "transactions": [
            {
              "header": {
                "batcher_pubkey": "02ebce312bde38c759781b5d1f1599875fdb9af3ad8f109a87aa6263e092ebb3a7",
                "dependencies": [],
                "family_name": "sawtooth_settings",
                "family_version": "1.0",
                "inputs": [
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c1c0cbf0fbcaf64c0b",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c12840f169a04216b7",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c1918142591ba4e8a7",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c12840f169a04216b7"
                ],
                "nonce": "",
                "outputs": [
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c1c0cbf0fbcaf64c0b",
                  "000000a87cb5eafdcc6a8cde0fb0dec1400c5ab274474a6aa82c12840f169a04216b7"
                ],
                "payload_encoding": "application/protobuf",
                "payload_sha512": "497fae61053b64811df34264c7b942e88efbf7931710ff4e9e53b519694e82c618aa30c26c1a55f2f6a319d951603c69c87a885e9d678e1c4224185532aa0ca4",
                "signer_pubkey": "02ebce312bde38c759781b5d1f1599875fdb9af3ad8f109a87aa6263e092ebb3a7"
              }
            ]
          }
        ]
      ]
    }
  ]
}

```

Figure 2.15 – Localhost blocks

After running this command, you should see a json object response with “data”, array of batches, header, etc.

After running these commands, we need to open new terminal and initialize json handlers for Hyperledger Sawtooth.

For this we need to move to following directory:

cd education/LFS171x/sawtooth-material/sawtooth-wada/processor

After we need to start npm (node packet manager). We use it to unpack containers.

```
nikita@VirtualBox:~$ cd education/LFS171x/sawtooth-material/sawtooth-tuna/processor
nikita@VirtualBox:~/education/LFS171x/sawtooth-material/sawtooth-tuna/processor$ npm start

> Sawtooth-tuna-chain-processor@0.0.0 start /home/nikita/education/LFS171x/sawtooth-material/sawtooth-tuna/processor
> node index.js

Initializing JSON handler for Sawtooth Tuna Chain
Connected to tcp://localhost:4004
Registration of [transfer-chain 0.0 application/json] succeeded
```

Figure 2.16 – Node packet manager start

After running all of these commands our system is ready to work. We need to open index.html file, which contains our blockchain application’s interface.

For this purpose we need to move to following directory:

cd education/LFS171x/sawtooth-material/sawtooth-wada/client/index.html.

Grab html file and paste it in the browser. You will see application’s interface on the screen.

Athlete name	Box number	Mission order	Time sealed	Doping Control Officer
--------------	------------	---------------	-------------	------------------------

Figure 2.17 – Interface of the blockchain application

With the help of this application we can transfer doping control forms to World Anti-Doping Agency with the help of blockchain.

Doping control officer creates public key and then enters:

- athletes name;
- box number of the sample;
- mission order;
- sealed time of the sample;

After pressing Create button he will see all the information on the ADAMS list below. ADAMS workers can easily log into application and see information about sample. The example is illustrated below.

The screenshot shows a web browser window with the address bar displaying a file path. The page has a green header with 'ADAMS' and 'World Anti-Doping Agency'. Below the header, there is a section for 'Doping Control Officer's public key' with a text input field containing a long alphanumeric string. Underneath, the 'Doping Control Form's components' section contains four text input fields: 'Mo Farah', 'N8956', 'N012365', and '14:08'. A blue 'Create' button is positioned below these fields. At the bottom, the 'ADAMS List' is displayed as a table with two rows of data.

Athlete name	Box number	Mission order	Time sealed	Doping Control Officer
Mo Farah	029af74e2dac5b755dd5af07791173c1f6e774333aad4e555d5a2d364065e81284			
Nikita Sharapov	029b65a187b3492c6417aff3140493e81d2b39ccee855ba8e4c45d7e9fa369e			

Figure 2.18 – Interface with current data

We have chosen following parameters to write down in our blockchain application:

- athlete name: Mo Farah;
- box number: N8956;
- mission order: N012365;
- time sealed: 14:08;

All of these parameters are transferred to blockchain structure and can be seen in ADAMS list of athletes. The process of adding athlete parameters into blockchain is illustrated below.

```

sawtooth-validator-default | 2018-06-04 13:14:46.154 DEBUG | Interconnect | ServerThread receiving CLIENT_BATCH_SUBMIT_REQUEST message: 923 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.137 DEBUG | Interconnect | ServerThread sending TP_PROCESS_REQUEST to b'babec827c-b367-44b8-96a2-e9815be72856'
sawtooth-validator-default | 2018-06-04 13:14:46.159 DEBUG | Interconnect | ServerThread receiving TP_STATE_GET_REQUEST message: 209 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.168 DEBUG | Interconnect | ServerThread sending TP_STATE_GET_RESPONSE to b'babec827c-b367-44b8-96a2-e9815be72856'
sawtooth-validator-default | 2018-06-04 13:14:46.168 DEBUG | Interconnect | TP_STATE_HANDLERS: GET: [{"19d83200ad019353b82e3a340beabdc990e2f559f144bd0e2efacd7e87a03ab991292", None}]
sawtooth-validator-default | 2018-06-04 13:14:46.168 DEBUG | Interconnect | ServerThread receiving TP_STATE_SET_REQUEST message: 312 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.168 DEBUG | Interconnect | TP_STATE_HANDLERS: SET: [{"19d83200ad019353b82e3a340beabdc990e2f559f144bd0e2efacd7e87a03ab991292"}]
sawtooth-validator-default | 2018-06-04 13:14:46.163 DEBUG | Interconnect | ServerThread sending TP_STATE_SET_RESPONSE to b'babec827c-b367-44b8-96a2-e9815be72856'
sawtooth-validator-default | 2018-06-04 13:14:46.165 DEBUG | Interconnect | ServerThread receiving TP_PROCESS_RESPONSE message: 71 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.165 DEBUG | Interconnect | Message round trip: TP_PROCESS_RESPONSE 0.008325815200805664
sawtooth-validator-default | 2018-06-04 13:14:46.266 INFO | publisher | Claimed block: ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.267 INFO | publisher | Block publishing is suspended until new chain head arrives.
sawtooth-validator-default | 2018-06-04 13:14:46.267 DEBUG | chain | Block received: ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.268 INFO | chain | Starting block validation of : ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.269 DEBUG | Interconnect | ServerThread sending TP_PROCESS_REQUEST to b'2a9b66a4-69f5-44ec-b320-f99ba87d0a63'
sawtooth-validator-default | 2018-06-04 13:14:46.272 DEBUG | Interconnect | ServerThread receiving TP_STATE_GET_REQUEST message: 209 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.272 DEBUG | Interconnect | TP_STATE_HANDLERS: GET: [{"19d83200ad019353b82e3a340beabdc990e2f559f144bd0e2efacd7e87a03ab991292", None}]
sawtooth-validator-default | 2018-06-04 13:14:46.273 DEBUG | Interconnect | ServerThread sending TP_STATE_GET_RESPONSE to b'2a9b66a4-69f5-44ec-b320-f99ba87d0a63'
sawtooth-validator-default | 2018-06-04 13:14:46.275 DEBUG | Interconnect | ServerThread receiving TP_STATE_SET_REQUEST message: 312 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.276 DEBUG | Interconnect | TP_STATE_HANDLERS: SET: [{"19d83200ad019353b82e3a340beabdc990e2f559f144bd0e2efacd7e87a03ab991292"}]
sawtooth-validator-default | 2018-06-04 13:14:46.276 DEBUG | Interconnect | ServerThread sending TP_STATE_SET_RESPONSE to b'2a9b66a4-69f5-44ec-b320-f99ba87d0a63'
sawtooth-validator-default | 2018-06-04 13:14:46.277 DEBUG | Interconnect | ServerThread receiving TP_PROCESS_RESPONSE message: 71 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.279 INFO | chain | on_block_validated: ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.280 INFO | chain | Chain head updated to: ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.288 INFO | publisher | Now building on top of block: ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.293 DEBUG | publisher | Loaded batch injectors: []
sawtooth-validator-default | 2018-06-04 13:14:46.294 DEBUG | chain | Verify descendant blocks: ae858321(2, 5:d3a400eb, P:9f2479fd) ([])
sawtooth-validator-default | 2018-06-04 13:14:46.294 DEBUG | state_delta_processor | Publishing state delta from ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.287 INFO | chain | Finished block validation of: ae858321(2, 5:d3a400eb, P:9f2479fd)
sawtooth-validator-default | 2018-06-04 13:14:46.288 INFO | Interconnect | ServerThread sending CLIENT_BATCH_SUBMIT_RESPONSE to b'a1c78585bd9c48be'
sawtooth-validator-default | 2018-06-04 13:14:46.294 DEBUG | Interconnect | Message round trip: TP_PROCESS_RESPONSE 0.0086226463178711
sawtooth-validator-default | 2018-06-04 13:14:46.295 DEBUG | Interconnect | ServerThread receiving CLIENT_STATE_LIST_REQUEST message: 81 bytes
sawtooth-validator-default | 2018-06-04 13:14:46.339 DEBUG | Interconnect | ServerThread sending CLIENT_STATE_LIST_RESPONSE to b'a1c78585bd9c48be'
sawtooth-validator-default | 2018-06-04 13:14:46.339 DEBUG | Interconnect | ServerThread receiving CLIENT_BATCH_SUBMIT_REQUEST message: 923 bytes

```

Figure 2.19 – Adding parameters into blockchain

Finally we have developed blockchain application for World Anti-Doping Agency (WADA) with the help of Hyperledger Sawtooth blockchain platform. Following tasks were done and in future this application can be implemented.

Transfer of authentic data used for the registration of an athlete and a sample in the international system ADAMS with the inability to implement subsequent changes in the registry was done fully. Errors in the process of filling in doping control form were corrected. Our database storage has the ability to look through transaction histories.

## 2.6 Calculation of network components

2.6.1 Reliability cluster. Readiness reflects the ability of the system to perform its functions continuously.

The availability factor is the probability that the computer system will be operational at any one time.

This coefficient is determined by the formula:

$$(2.1) \quad K = \frac{MTBF}{MTBF + MTTR}$$

Where:

MTBF means Mean Time Between Failure

MTTR – Mean Time To Repair

Unlike reliability, the value of which is determined only by the value of MTBF, the availability also depends on the time required to return the system to operational state.

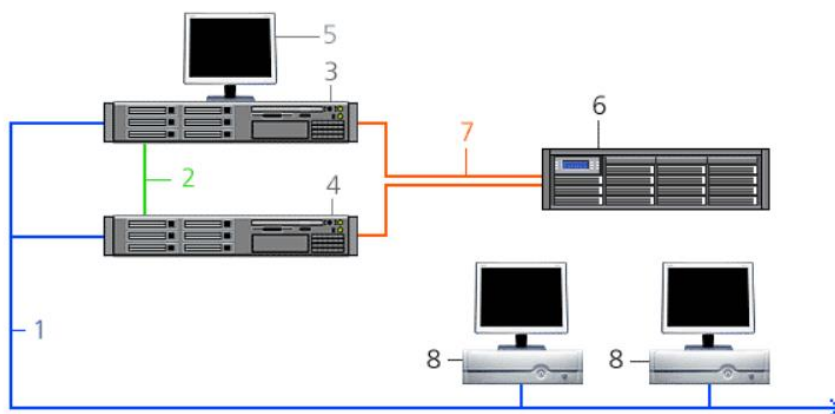


Figure 2.20 – Cluster structure

Where 1 – Public network; 2 – Private network; 3 – Node 1; 4 – Node 2; 5 – Management console; 6 - Common disk array; 7 – SCSI and Fiber Channel interfaces; 8 – Clients

The cluster consists of two nodes (servers) connected to a common disk array. All the main components of this disk array - power supply, disk drives, I / O controller - have hot-swappable redundancy. An internal network to exchange information about their current state interconnects the cluster nodes. The cluster is powered by two independent sources. The connection of each node to an external local network is also duplicated.

Thus, all subsystems of the cluster have redundancy, therefore, if any element fails, the cluster as a whole will remain operational. Moreover, replacement of a failed element is possible without stopping the cluster.

Therefore we pretend that our system has also two nodes (servers) connected to a common disk array. Actually this number may differ because we are building blockchain application with dozen of nodes.

The application, together with its resource group, is not bound "hard" to the specific node, but, on the contrary, can be started on any of these nodes (and several applications can run simultaneously on each node). In turn, the clients of this application (service) will "see" not the nodes of the cluster on the network, but the virtual server (the network name and IP address) on which the application is running.

First, the application runs on one of the nodes. If this node for some reason ceases to function, the other node ceases to receive from it an activity signal ("heartbeat") and automatically starts all applications of the failed node, i.e. The applications, together with their resource groups, "migrate" to a working node. Migration of the application can last from several seconds to several tens of seconds and during this time this application is not available to clients.

Depending on the type of application after the restart, the session resumes automatically or the client may need to be reauthorized. There is no need to change the settings on the client side. After repairing a faulty node, its applications can migrate back.

As we are building high availability cluster we need to take into account the fact that this system can break down. There is no system in the world that is fully for 100% protected from faults.

If we need to do some stops we can do it easily without necessity of stopping all the system.

On a cluster, these operations can be performed sequentially on different nodes, without interrupting the operation of the cluster as a whole.

Unplanned shutdowns occur due to software or hardware failures. In the event of a software failure on a normal server, you will need to reboot the operating system or application, in the case of the cluster, the application will migrate to another node and continue working.

Anyway, we need to have some numbers on which to rely on while constructing this system.

As we saw from above formula MTBF means Mean Time Between Failures. How can we now this value? There is a special formula for it:

$$MTBF = \frac{\text{Testing time} * \text{Number of products tested}}{\text{Number of failed products}} \quad (2.2)$$

Let's pretend that for our blockchain system:

Testing time = 5 years;

Number of products tested = 5000 items;

Number of failed products = 500 items

$$MTBF = \frac{5 * 5000}{500} = 50 \text{ (years)}$$

This means that only in 50 years all items will be broke down.

The concept of MTBF does not reflect at all what obviously follows from its name. "Mean time between failures" literally means time, which is only half the MTBF. So, in our example this "average time" will not be 50 years, but only 25 years, because on average all copies of the product will not work for 10 years, but half less. Those. MTBF, claimed by the manufacturer - this is the time during which the product will fail with a probability of 100% [31].

Therefore, we can pretend that probability of MTBF failure equals to 1 and if we'll measure MTBF in years, then he probability of component failure during one year will be:

$$P = \frac{1}{MTBF} = \frac{1}{50} = 0,02$$

For example we need to replace broken component of the system within 24 hours (1/365 of the year) then probability of this occasion is equaled to:

$$P_d = \frac{P * P}{365} * 2$$

There are two different possibilities:

1. Component # 1 failure at any point in time during the year (probability P)
2. The failure of component # 2 within 24 hours after the release of component #1 (probability  $P / 365$ )

The probability of failure-free operation of any component during the year equals to:

$$P'_i = 1 - P_i \quad (2.3)$$

where  $P_i$  - probability of component failure within one year

Usually lifetime of the motherboard is a year, in some other cases - 10 years. However, the average life of the motherboard is usually 5-6 years. Much depends on the intensity of this board, and if you operate the computer sparingly, the motherboard will be able to work even 15 years.

HDD (hard disk) is one of the weakest in terms of survivability. Modern models are designed to work from three years, although in practice they last longer. It is noticed that older models were hardier, and modern disks fail even after 5 years.

Most of the energy consumed by power source goes to the power supply of the components, and not to the heating of the unit itself. Therefore, expensive reliable models may not even have a fan, since it unnecessary there. The life expectancy of a computer power supply of this type can be 10-20 years. So power source with fan works approximately for 10-20 years, therefore we can conclude that fan works for 10-20 years too.

CPU's lifetime is 10 years; however, if it is still running and is capable of adjusting voltages and frequencies, then by implementing an effective heat removal system from the system unit, you can extend the processor life to 50 years. Anyway, let's take 10 years.

On average, the life of the video card in normal operation is from 3 to 8 years. But it all depends on the degree of congestion of the device. If you constantly play in games or even earn on it a crypto currency, then you cannot count on a long period of work. Such cards will work for 2-3 years [13].

RAM (random access memory) is eternal. There's practically nothing to break. However, it can fail in the event of a high temperature inside the system unit or the supply of an incorrect voltage to it (this is when the motherboard fails). Also, the bar can physically break if touched. Therefore let's take 100 years for RAM.

The probability of failure-free operation of any component during the year is:

$$P'_m = 1 - P_m = 1 - \frac{10}{365} = 1 - 0,0273 = 0,9726$$

$$P'_{HDD} = 1 - P_{HDD} = 1 - \frac{5}{365} = 1 - 0,0136 = 0,9863$$

$$P'_{ps} = 1 - P_{ps} = 1 - \frac{15}{365} = 1 - 0,0411 = 0,9589$$

$$P'_{v} = 1 - P_{v} = 1 - \frac{20}{365} = 1 - 0,0547 = 0,9452$$

$$P'_{CPU} = 1 - P_{CPU} = 1 - \frac{10}{365} = 1 - 0,0273 = 0,9726$$

$$P'_{vc} = 1 - P_{vc} = 1 - \frac{7}{365} = 1 - 0,0191 = 0,9801$$

$$P'_{RAM} = 1 - P_{RAM} = 1 - \frac{0,1}{365} = 1 - 0,00002 = 0,9999$$

The probability of failure-free operation of all components during the year is equal to the product of the probabilities of these independent events equals to:

$$(2.4) \quad P_{s'} = P'_m * P'_{HDD} * P'_{ps} * P'_v * P'_{CPU} * P'_{vc} * P'_{RAM}$$

$$P_{s'} = 0,9726 * 0,9863 * 0,9589 * 0,9452 * 0,9726 * 0,9801 * 0,9999 = 0,8287$$

Then the probability of server failure during the year equals to:

$$P_s = 1 - P_{s'} = 1 - 0,8287 = 0,1713$$

Knowing the probability of server failure within a year, you can determine the time it takes to fail (the time through which the server will fail with a 100% probability):

$$MTBF_s = \frac{1}{P_s} = \frac{1}{0,1713} = 5,8377 \text{ (years)}$$

Now we can determine the availability coefficient:

$$K_s = \frac{MTBF_s}{MTBF_s + MTTR_s} = \frac{5,8377}{5,8377 + \frac{1}{365}} = \frac{5,8377}{5,8404} = 0,9995$$

Let us summarize the manufacturers' data on the reliability of individual components in the following table.

Table 2.1 – Reliability of individual components

Server components	MTBF (hours)	MTBF (years)	Probability refusal during the year	Amount of components in the server	Probability refusal with duplication
Motherboard	87 600	10	0,0273	1	0,0273
HDD	43 800	5	0,0136	1	0,0136
Power source	131 400	15	0,0411	2	0,0016
Ventilator	175 200	20	0,0547	2	0,0029
CPU	87 600	10	0,0273	1	0,0273
Video card	61 320	7	0,0191	1	0,0191
RAM	876 000	100	0,00002	1	0,00002
Total			0,18312		0,09182

Therefore, we obtain following results:

- the probability of server failure during the year – 0,09182
- MTBF of the server – 5,8377 (years)
- average time of elimination of malfunction – 24 (hours)
- server availability – 99,9 %
- average idle time per year: 3,65 (hours)

2.6.2 Block size mathematics. Currently there is a hard limit for 1 MB block size; all major Bitcoin clients, regardless of blockage 1, are considered invalid [3]. This limit is estimated at 4,000 transactions per block (assuming the average transaction size is the most relevant, approximately 200-250 bytes). Since blocks are mined once every 10 minutes, there are approximately 7 transactions per second (TPS). The enhanced behavior gives the maximum processing time for actual ideas in the queue, and some studies have estimated that they have never been limited by 3.5 TPS [4].

Mathematical modelling shows that has increased as much pressure as possible, for example 2.8 transactions per second (up to 80%). The negative impact of the confirmed transaction time on the network is negative. The confirmation time of the first transaction is 18.5 minutes; half of the transaction speed has been confirmed. To provide a comparison, if the input is 1 TPS, the first confirmed central confirmation time is 7 minutes.

The building of a mathematical model is actually the main stage of design or research of any method. The whole consequent analysis of the object is dependent on the quality of the model. Creating a model isn't the proper process. Highly is determined by the researcher, his taste and experience, usually depends on a particular experimental material. The unit must be completely precise, ample and ought to be handy for usage.

Table 2.2 Processing time of transactions depending on block size and network load

Throughput, tps	Block size, MB	Network load, %	Median processing time, min	Processing of 90% tx, min
1,75 (normal)	1	50.0	8.5	29.0
	2	25.0	7.0	23.5
	3	16.7	7.0	23.0
	4	12.5	7.0	23.0
	8	6.3	7.0	23.0
	20	2.5	7.0	23.0
2,5 (peak)	1	70.0	13.2	42.8
	2	35.0	7.4	24.7
	3	23.3	7.0	23.0
	4	17.5	7.0	23.0
	8	8.8	7.0	23.0
	20	3.5	7.0	23.0
3,5 (maximum)	1	100.0	129.1	380.0
	2	50.0	8.5	29.0
	3	33.3	7.4	24.7
	4	25.0	7.0	23.5
	8	12.5	7.0	23.0
	20	5.0	7.0	23.0

Throughput is another aspect of the problem: If the parameter is a confirmation that all the dust in the transaction can be slowed down most of the time - than the transaction, the output value of the average transaction cost is even lower. [7]

Dust is usually used to service (DOS) objections, which was the largest in July 2015 [2]. Certain types of pole transactions are not broadcast at very low cost and are not included in the block. More restrictions on trading: as an alternative to stopping growth. The scale of the block comparison grows, and the behavior is an obligation of interest: to make the yarn soft (that is, to be accepted by the minerals most advanced software upgrade block software) if hard to enlarge the block size (ie, not correspond to changes agreement).

Generally, the supreme objective of an assailant is usually to totally prevent the internet resource - "denial of service". An assailant also can demand cash to stop an assault. In certain instances, a DDoS attack might be an attempt to discredit and / or ruin a competitor's company [16].

To send an extremely big selection of requests to the victim 's resource, a cybercriminal typically produces a network of infected "zombie computers." Since the perpetrator regulates the activities of each infected personal computer in a zombie networking, the strike may be way too effective for the victim's net useful resource.

Table 2.3 Resource consumption by full nodes as the block size increases

Characteristics	Scale factor	Block size, MB ( $=N/2$ )						
		0,5	1	2	4	8	16	32
Transaction throughput, tps	N	1.75	3.5	7	14	28	56	112
Number of txs in a block	N	1050	2100	4200	8400	16800	33600	67200
Blockchain storage per day, MB	N	72	144	288	576	1152	2304	4608
Blockchain storage per year, GB	N	26	51	103	205	411	821	1643
Transaction processing time, ms	1	0.33	0.33	0.33	0.33	0.33	0.33	0.33
Block verification time, s	$N + 0.09N \log_2 N$	0.07	0.15	0.33	0.71	1.51	3.23	6.86
Average bandwidth, kB/s	N	74	148	296	592	1184	2368	4736
Daily traffic, GB	N	6.2	12.4	24.8	49.6	99.2	198	397
Yearly traffic, TB	N	2.2	4.4	8.8	17.7	35.4	70.7	141
RAM usage, GB	N	2	4	8	16	32	64	128
Immediately excluded nodes, %	n/a	0	20	40	75	90	95	95
Excluded nodes in 6 months, %	n/a	5	25	50	80	95	95	95

Increasing the block size could lead to additional strain on the nodes of the Bitcoin network. The most computationally expensive operation during transaction relay is verification of the ECDSA signatures of the transaction inputs (each transaction is verified before it is relayed to other nodes in order to protect the network from DoS attacks). Modern CPUs are able to verify several thousand transactions per second [11]. Thus, current transaction throughput of several tps is not much of a burden for the network.

I have calculated parameters for resource consumption and have taken average values of block size which is currently 0.5 MB.

Transaction throughput is block size divided by the expected time interval between blocks (600 seconds) and by the current average transaction size (slightly less than 0.5 kilobytes)

This methods yield a result of approximately 1:75 tps.

Number of transactions in a block is the block size divided by the average transaction size; alternatively, it is equal to transaction throughput times 600.

$$\text{Transaction in a block} = \text{Thransaction ththroughput} * 600 = 1050$$

Blockchain storage is a mean block size multiplied by the expected number of blocks. 144 blocks per day;  $144 * 365 = 52560$  blocks per year.

$$(2.3) \quad \frac{\text{Blockchain storage}}{\text{day}} = \text{block size} * \frac{\text{blocks}}{\text{day}}$$

$$\frac{\text{Blockchain storage}}{\text{day}} = 0.5 * 144 = 72 \text{ MB}$$

$$(2.4) \quad \frac{\text{Blockchain storage}}{\text{year}} = \text{block size} * \frac{\text{blocks}}{\text{day}} * \text{days in a year}$$

$$\frac{\text{Blockchain storage}}{\text{year}} = 0.5 * 144 * 365 = 26280 \text{ MB}$$

Transaction processing time is based on an assumption that a node can process 3000 transactions per second [11]. Note that each transaction needs to be parsed, then the node needs to lookup the unspent transaction outputs corresponding to transaction inputs and check each of signatures embedded in inputs.

Block verification time is transaction processing time multiplied by the mean number of transactions in a block times 0.2. The last factor corresponds to the notion that most transactions in a block are already verified when the block arrives; node only needs to calculate transaction hashes and look them up in the transaction cache.

$$\text{Block verification time} = 0.2 * 1050 * 0.0003333 = 0.07 \text{ sec.}$$

According to statoshi.info average bandwidth of a node is 74 kbs. (13)

Currently a node processes more that 6 gigabytes (GB) of traffic per day (13)

RAM usage is the most difficult variable to estimate as it depends on many factors. Furthermore, RAM usage differs for various types of nodes. One of the major components contributing to RAM consumption is the unspent transaction output set, which currently takes more than 4 GB [14]. The set does not have to reside completely in RAM, although other storage methods lead to elevated transaction verification time. Let's use an empirical value of 2 GB based on recommendations [15] which is lower than what other measurements suggest [16].

Obviously, for specialized nodes such as mining nodes, the requirements on RAM are higher, as these nodes need to store the entire unspent transaction output set in RAM in order to rapidly verify incoming transactions and blocks.

Let's calculate key node parameter estimates for increased block sizes by multiplying our baseline values by scaling factors in the second column of the table, with  $N$  denoting the multiplier for block size. These scaling factors are established as follows. We assume that the average size of a transaction remains the same, so transaction throughput and the number of transactions per block scale linearly with block size. Similarly, it is obvious that blockchain storage linearly depends on the block size as well. Transaction processing requires searching in the unspent transaction output set. With an optimal implementation, average search time depends logarithmically on a size of the set; the latter should scale linearly (as it is implied by historic data [17]). As the number of unspent transaction outputs is currently well over 107, transaction processing time would remain nearly the same as the block size increases.

Block verification time exhibits the fastest growth rate as the block size increases; while most transactions in a block should already be verified when it arrives, a node still has to compute hashes for all transactions and look them up in the cache. As in the previous case, lookup times scale logarithmically depending on the size of the cache.

Let  $S$  denote the current size of the transaction cache and  $t$  denote average time to look up a single transaction. If the block size increases  $N$  times, time to look up each transaction increases to  $t * \log_2(NS)$ ; thus, the block verification time would increase by a factor of

$$(2.6) \quad N \frac{t * \log_2(NS)}{t * \log_2 S} = N \left( 1 + \frac{\log_2 N}{\log_2 S} \right)$$

According to statoshi.info,  $S = 2000$  and suppose that block increases  $N = 0.1$  times by increasing block size twice which implies the scaling factor:

$$\text{Block verification time} = 0.1 \left( 1 + \frac{\log_2 0.1}{\log_2 2000} \right) = 0.07 \text{ sec}$$

The transport node, and the size of the block that depends on the transaction throughput. Both of these values are the same.

We believe that so far, the node may keep the same percentage as the SAP transaction output; because the collection chain increases with the size of the block, RAM can also be used all the time.

Other components that help reduce costs (such as behavioral cache size) show linear or sublinear growth and therefore cannot be reduced from long-set setup behavior. Like other components, RAM usage is primarily accomplished through user settings; a node can run relatively low RAM, but it may delay the transaction.

In the table, since the common block size improves, the amount of nodes which have no hardware update in the table is within the table. These assumptions think that a lot of customers run complete nodes on customer level hardware and then run on an individual pc or perhaps cloud. The node hardware is recognized by

steam measurement [19]; we think that computer game players as well as Bitcoin enthusiasts have exactly the same quantity of information focused on the hardware of theirs. RAM is actually an exception: we believe the node supporting the personal computer is actually under 3 GB and that it takes no less than 2 GB of RAM to work with the margin as the node [15]. For instance, in case the block size increases to 2 MB, the node must devote 8 GB of RAM to the Bitcoin prospect, what about the survey, much more than one half of the PCs are actually under RAM.

We likewise feature an estimation of exactly how current node networks will likely be removed in the following 6 months. The primary decrease in the selection of nodes is primarily associated with the usage of CPU and RAM, and also in the end there are additional restricted things including disk space as well as Internet traffic. For instance, in the situation of an ordinary block size of 8 MB, the node must have greater than 34 GB of disk capability to deal with roughly 3 TB (TB) of visitors every month.

### **3 Life safety activity section**

#### **3.1 Analysis of working conditions**

This graduation project is devoted to the development of the blockbuster platform on the Hyperledger operating system. The application will be developed in

A private room, and due to this, it is necessary to determine the necessary working conditions and methods of preventing harmful factors.

This diploma project is devoted to the development of a program for hiding information in images. The application is developed in a private room, and due to this it is necessary to determine the necessary working conditions and prevent harmful factors.

Safety of life activity is built thanks to these tasks:

- identifying the type of threat through quantitative characteristics and location;
- security, the basis for ensuring a cost-benefit ratio;
- reduction of possible hazards, based on the concentration of residual risk, as well as reducing the effects of hazards on humans.

The purpose of this work is to study the parameters for ensuring safety in the workplace.

The main tasks of this work are:

- 1) identify factors that indicate the safety of workers' health;
- 2) examine organizational factors; c) calculate the necessary calculations.

Development of the diploma project "Analysis of the principles of the construction of algorithms, data structures of blockchain technology" was conducted in the premises in accordance with the provisions that fall into the category of premises without increased danger. The temperature in the premises is maintained at the proper level of  $23 \pm 3$  ° C, humidity  $51 \pm 11\%$ , air movement 0,2 m / s, without the conditions listed in the GOST: humidity or current-conducting dust; conductive floors; high temperature; the possibility of simultaneous contact

with the ground of the metalwork of the building; having contact with the earth's surface; on the one hand, and with the metal cases of electrical equipment.

To ensure the necessary parameters of the working environment in the hall, the following requirements are met:

- the walls of the premises are painted with light paint, which excludes the settling of dust on them, with a color that does not tear the eyesight;
- technical floors in the room are covered with linoleum, which allows for convenient, quick and high-quality cleaning of dust and washing of floors;
- to minimize the effect of psychological factors, there are periodic breaks in working with the PC, but not more than 20 minutes. With an 8-hour shift, with a lunch break after 4 hours of work, for more regulated breaks, 3 hours after starting work and 2 hours before the end .

The mode of labor and rest for operators working directly with the PC depends on the nature of the work: when entering data, the information is read from the screen. Continuous work does not exceed 4 hours, 8 hours a day, every hour of work break for 5-10 minutes and once in 2 hours break for 15 minutes.

The number of working hours per week should not exceed 40. The main task of the operator is to obtain and enter information, monitor and correct settlement tasks on the PC using software and timely actions of a malfunction or network error [12].

Due to the fact that the work is carried out by the operator constantly in a sitting position, improper installation of equipment and its uncomfortable workplaces entails undesirable physiological changes, fatigue, and, as a result, an increase in industrial injuries.

An important role is played by the workplace plan. When organizing a workplace plan, it is necessary to take into account the data of human anthropometry. A person without much effort should have access to all documents, objects and devices located in his work area. Furniture is designed to give unhindered access to any part of the sanitation room for cleaning the premises.

Requirements for the room include the requirements for installing a computer and the requirements for maintaining the necessary parameters of the microclimate:

- the room should maintain the average temperature, there should be no dust and low relative humidity;
- The size of the room must satisfy the main sanitary and technical requirements for the location of the main equipment, and also provide space for cables, passageways, technical repair and maintenance of equipment. One workplace should account for not less than 15 m of the working space and not less than 4.5 m<sup>2</sup> of the area;
- the noise level in the workplace should not exceed the established standards.

The norms of temperature, relative humidity and speed of air movement in the working area (Figure 3.1).

The placement of technical equipment and the operator's chair in the work area provides: easy access to the main functional blocks and equipment, the

exclusion of accidental activation of control and information input, the most optimal placement of the operator's equipment is shown in Figure 3.1 [21]

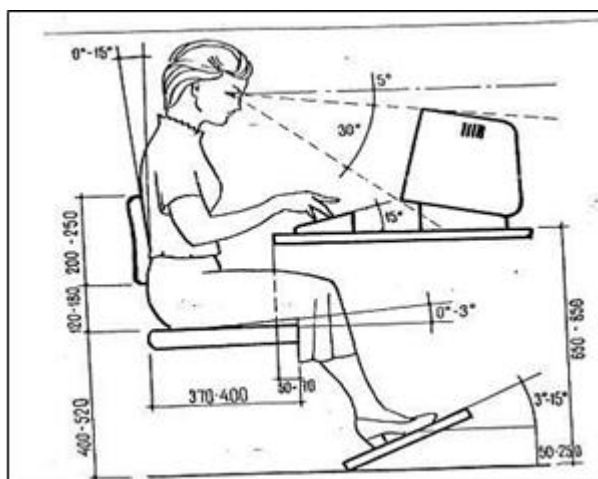


Figure 3.1 - Optimal characteristics of the operator's workplace

The workroom in question is located in a building that is far from railroad tracks or loaded motorways, airports, etc., so there are no external sources of noise that affect the work process. The noise level corresponds to the norm in accordance with GOST12.1.003-83 SSBT. Noise. General safety requirements ".

The building belongs to the I degree of fire resistance (Buildings with load-bearing and enclosing structures of natural or artificial materials, concrete or reinforced concrete with the use of sheet incombustible materials). The fire safety room belongs to the category "D" (reduced fire hazard in accordance with the TC of the RK "General requirements for fire safety"). In accordance with the standard rules of fire safety, administrative buildings and individual premises, and technological installations are provided with primary fire extinguishing means in accordance with the standards.

3.1.1 Layout of the room. The room has the following parameters:

- is located on the third floor of a five-story building;
- room (room) dimensions: length 8 m, width 3 m, height 3 m;
- type of light transmitting material - sheet glass, double; - type of binding - steel, double, opening;
- sunscreens - retractable adjustable shutters;
- two windows measuring 1.5 \* 1;
- interior walls - light;
- the room according to the visual conditions of work belongs to the category of light work (light physical, category Ia, the work is done sitting and does not require physical stress);
- artificial lighting - 2 lamps with two fluorescent lamps.

Figure 3.2 shows the layout of the room, where 1 is the window, 2 is the workplace, 3 is the door, and 4 is the air conditioner.

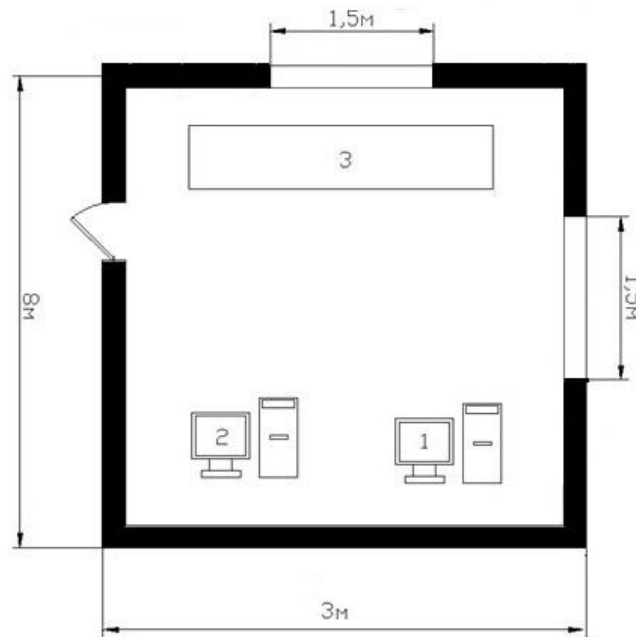


Figure 3.2 - Room plan

Characteristics of the equipment used in the work:

- laptop Asus UX430UQ, Intel Core i7 7 GEN, 8 Gb DDR4, 512 SSD, Nvidia GeForce 940MX;
- power supply: alternating voltage 220 - 250 V, frequency 50 Hz and power 90 W and 65 W respectively;

Electrical equipment is a potential source of fire hazard. Low noise equipment - there is no harmfulness as an increased noise.

3.1.2 Electrical safety. When working with power tools, you must follow the following rules:

- do not work with power tools in any form of intoxication;
- never leave power tools unattended;
- do not bring water into the room because of the danger of water ingress on bare wires and connectors;
- do not use tools, power filters, or other electrical devices with damaged rubber winding wires.

The building must be equipped with secure networked devices. Each room should be connected by its automatic machine, which in automatic mode, with a direct closure or exceeding the permissible load, will disconnect the sockets in the room from the network.

3.1.3 Fire safety. Fire safety means are represented by two sensors, catching smoke, connected to the general network of sensors of the University, as well as a serviceable fire extinguisher corresponding to GOST. [5] Safety in the laboratory is described in the safety rules of the mechatronics and information systems laboratory.

All members of the laboratory, university staff who have access to the laboratory, students who have obtained permission to work in the laboratory, have been acquainted with safety technology.

The laboratory complies with the fire safety requirements described in the technical regulations "General requirements for fire safety". [6]

Questions of fire safety are described in the "Fire Safety Rules of the Republic of Kazakhstan" on October 9, 2014 No. 1077. [14]

This technical regulation is chosen because it defines the main provisions of technical regulation in the field of fire safety and establishes:

Classification of fires and their hazardous factors, substances and materials, as well as technological environments for explosion and fire hazards; explosive and fire hazardous areas; building materials for fire hazard; building structures and fire barriers; electrical equipment, outdoor installations, buildings, structures and premises for explosion and fire hazard, which is given in Annex 1 to this Technical Regulations;

Fire safety requirements for business objects of various purposes at all stages of their life cycle; when designing urban and rural settlements; design and construction of buildings and structures; to production.

Fire safety is provided by means of the following systems:

1. prevention of fire;
2. fire protection;
3. organizational and technical measures.

During the implementation of the project, one of the important tasks is to ensure fire safety of the premises, as well as the safety of people in this room. To perform these tasks, special attention is needed to the following points:

1. fulfillment of requirements for fire safety, and special attention to flammable objects, established by technical regulations, regulatory legal acts of the Republic of Kazakhstan and regulatory documents governing fire safety;
2. the risk of fire should be within the limits established by the technical regulations.

Since, fire safety is ensured by a number of systems, it is necessary to check them in order.

### **3.2 Calculation of the natural lighting of the room**

I calculate the area of the side light apertures in the room, which is necessary to create a normalized illumination in the workplace.

The room has the following dimensions: length  $L = 8$  m, width  $B = 3$  m, height  $H = 3$  m.

The height of the working surface above the floor level,  $h_p = 0,8$  m, the window starts from the height  $h_1, h_2 = 0,8$  m, the height of the window,  $h_o = 1,5$  m. The working room is located in the IV time zone - in Almaty ( belt of light climate - IV 500 north latitude and south (Almaty) According to SNiP RK 2.04.-05.2002 [20, 21]  $e_H = 1,2$  %.

Calculation of natural light is to determine the area of light apertures. With side lighting (light apertures in the exterior walls of the building), the area of the light apertures  $S_0$ , which provides the normalized values of KEO, can be determined proceeding from the relation 4.2.

$$100 \frac{S_0}{S_n} = \frac{e_n K_3 n_0}{t_0 r_1} K_{3Д} \quad (4.1)$$

where:

$S_f$  - floor area of the room, m<sup>2</sup>;

$S_0$  - the area of the light apertures, with side illumination of m<sup>2</sup>;

$k_3$  - safety factor [20];

$\eta_0$  - light characteristics of windows [20];

$K_{3Д}$  - is a coefficient that takes into account the darkening of windows by opposing buildings [20];

$r_1$  - is the coefficient that takes into account the increase in KEO with side lighting, due to light reflected from the surface of the room and the underlying layer adjacent to the building [20];

$\tau_0$  - is the total light transmittance;

From the relation, we obtain a formula for determining the area of the light apertures of  $S_0$ .

$$S_0 = \frac{e * n_0}{100 * t_0 * r_1} * k_{zd} * k_z * S_n \quad (4.2)$$

where:

$S_f$  - is the floor area of the room (n<sup>2</sup>),

$e_n$  - is the normalized value of KEO,

$k_3$  - is the safety factor

$\eta_0$  - light characteristics of windows,

$r_1$  - is a coefficient that takes into account the increase in KEO at the side Lighting due to light reflected from the surfaces of the room and the underlying layer adjacent to the building,

$k_{3Д}$  - factor, taking into account the shading of windows with opposing buildings.

Floor area of the room

$$S = L * B = 8 * 3 = 24 \text{ m}^2$$

The total light transmittance is calculated by the formula 5.4

where:

$\tau_1$  - coefficient of light transmittance of the material according to Table 6 [1]:  
for double glass

$\tau_2$  - coefficient that takes into account the loss of light in the light-coverings according to Table 7 [1]:

$\tau_3$  - the coefficient that takes into account the loss of light in the bearing structures, with side illumination;

$\tau_4$  - coefficient that takes into account the loss of light in sunscreens, see table 3.6 [1]:

Since the window glass is double, then  $\tau_1 = 0,8$ . For steel forms of the supporting structure  $\tau_2 = 0,75$ . For retractable adjustable blinds and curtains  $\tau_3 = 0,9$ . For retractable adjustable blinds and curtains  $\tau_4 = 1$ .

Substituting the values, we obtain

$$\tau_0 = 0,8 * 0,75 * 0,9 * 0,9 = 0,486$$

The area of the room is  $S = 24 \text{ m}^2$ . The stock factor for classrooms, classrooms and trading rooms is  $K_3 = 1,2$ . Proceeding from the ratio of the length and depth of the room, as well as the depth of the room to its height from the level of the conventional working surface to the top of the window  $\eta_0 = 8$ . The factor that takes into account the darkening of windows by the opposing buildings  $K_3 \text{ Д} = 1,2$ . The coefficient that takes into account the increase in KEO in side lighting, Taking factor  $\rho_{av} = 0,5$ , the ratio  $r_1$  is equal to  $r_1 = 2$ .

Substituting the values in the formula we get:

$$S_0 = \frac{1,2 * 8}{100 * 0,486 * 2} * 1,2 * 1,2 * 24 = 3,4 \text{ m}^2$$

Conclusion: The lighting parameters of the operator's room were determined using the coefficient of use, designed to calculate the overall uniform illumination of horizontal surfaces in the absence of large shading objects. The essence of this method is to determine the value of the coefficient  $\eta$ , equal to the ratio of the light flux incident on the calculated surface, to the total flow of the lighting device.

### 3.3 Calculation of artificial lighting of the room

Length of the room  $L=8 \text{ m}$ , width of the room  $B=3 \text{ m}$ , height is  $H=3 \text{ m}$ . The height of the working surface above the floor level is  $h_p=0,8 - 1 \text{ m}$ . The distance from the luminaire to the ceiling is  $h_c=0 \text{ m}$ . Windows with retractable external adjustable jalousie, with insulating floor (linoleum), proceeding from this we take the coefficients of reflection of the ceiling, walls and floor:  $\rho_{\text{пот}} = 70\%$ ,  $\rho_{\text{ст}} = 50\%$ ,  $\rho_{\text{пол}} = 30\%$ . The height of the working surface is  $H_p = 0,7 \text{ m}$ ;

Calculation of lighting using the coefficient of utilization. This method consists in determining the value of the coefficient  $\eta$  equal to the ratio of the light flux incident on the calculated surface to the total flux of the lighting device. The value of the coefficient  $\eta$  is found from the tables linking the geometric

parameters of the premises (the index of rooms  $i$ ) with their optical characteristics (the reflection coefficients of the ceiling  $p_{\text{пот}}$ , the walls  $p_{\text{ст}}$  and of the floor  $p_{\text{п}}$ ). The index of the room  $i$  is determined by the formula.

$$i = \frac{A \cdot B}{h \cdot (A+B)} \quad (4.3)$$

where  $L$  - the length of the room,  $B$  - width of the room;  
Calculate the calculated height of the suspension by the formula:

$$h_p = H - h_{\text{пн}} - h_{\text{св}} \quad (4.4)$$

where  $H$  – the height of the room;  
 $h_{\text{пн}}$  – height of the working surface;  
 $h_{\text{св}}$  – the height of the overhang of the luminaire

$$h_p = 3 - 0,8 - 0 = 2,2 \text{ (m)}$$

Define the distance between the rows of lamps according to the formula:

$$L_A = \lambda \cdot h_p, \text{ m}; L_B = \lambda \cdot h_p$$

$$L_A = \lambda \cdot h_p = (0,6 \div 2) \cdot 2,2 = 1 \cdot 2,2 = 2,2 \text{ (m)}.$$

$$L_B = \lambda \cdot h_p = (0,6 \div 2) \cdot 2,2 = 0,7 \cdot 2,2 = 1,54 \text{ (m)}$$

Define the index of the room:

$$i = \frac{24}{2 \cdot (8 + 3)} = 1,1.$$

Define the required number of luminaires using the formula :

$$N = \frac{E \cdot K_3 \cdot S \cdot Z}{n \cdot F_l \cdot \eta} \quad (4.5)$$

where  $K_3$  the safety factor is determined taking into account the type of premises and the use of gas-discharge lamps:  $K_3 = 1,2$ . [20].

$z = 1,1 \div 1,2$  – coefficient of uneven lighting;

$n = 3$  – number of lamps in the luminaire;;

$\eta = 51\% = 0,51$  – coefficient of use of the light flux

$F_l = 3200 \text{ lm}$  – light flux

$E_{\text{min}} = 300 \text{ lux}$  – normalized illumination for artificial lighting

For the values of reflection, it is taken

$$\rho_c = 70\%, \rho_w = 50\%, \rho_f = 30\%.$$

Defining coefficient  $\eta$ , which is equal to  $\eta = 51\%$ .

For lighting of laboratory room we will use luminescent gas-discharge lamps with power of 40 W and a nominal light flux of 3200 lm. As luminaires we will use luminaires of PVLМ 1x40 type.

$$N = \frac{300 \cdot 1,2 \cdot 24 \cdot 1,2}{3 \cdot 3200 \cdot 0,51} = 2 \text{ lamps}$$

The actual illumination is determined by formula (3.8)

$$F = \frac{E \cdot K_r \cdot S \cdot Z}{N \cdot \eta} \quad (4.6)$$

$$F = \frac{300 \cdot 1,2 \cdot 24 \cdot 1,1}{2 \cdot 0,51} = 3105 \text{ lm.}$$

Define the distance from the lamp to one and up to another wall:

$$l_a = (0,4 \div 0,5) \cdot L_A = 0,5 \cdot 2,2 = 1,1 \text{ (m)}$$

$$l_B = (0,4 \div 0,5) \cdot L_B = 0,5 \cdot 1,54 = 0,77 \text{ (m)}$$

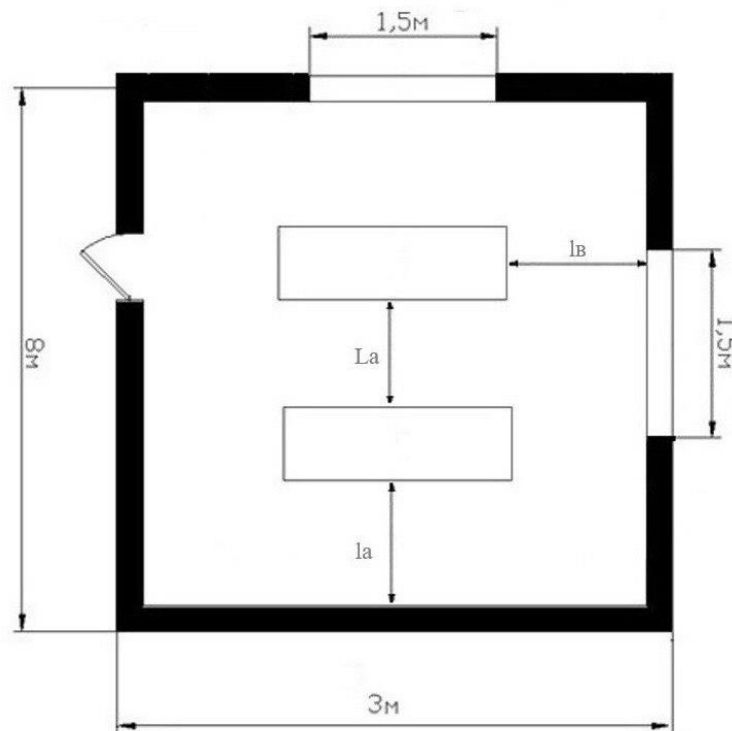


Figure 3.3 – Plan of arrangement of luminaries

Lamps we arrange in 2 rows with 1 lamp in each row. The distance from the wall to the first and last row of luminaires is 1.5 m, between two adjacent rows - 2

m. In the row, the luminaires are located at a distance of 0.5 m, and the outer edge of the first and last luminaire in the row lags by 0,5 m.

Conclusion for the life safety part. In order to create the most favorable working conditions in the room, artificial lighting is used to provide the normalized value of KEO in conjunction with natural light. Natural lighting does not properly provide, during the whole working time, the necessary lighting due to changes in the time of the day, or changes in the weather, and for this purpose a lighting system consisting of lamps with lamps is provided.

To create normalized illumination 2 lamps with each in 1 row, the power of each lamp should be not less than 40 W, which corresponds to the reality, and therefore the available illumination is sufficient.

The normalized values of noise levels at the workplace were also studied. Excessive noise causes hearing damage. Loud sounds can cause hearing loss either immediately or gradually over a long period of time. Damage can be immediately caused by the impact of peak sound waves caused by explosive sounds, such as shooting, explosions or instruments controlled by cartridges. Anyone can be exposed to excessive noise. Those who work in noisy workplaces, factories, foundries, work with power tools, plants and machinery and in noisy environments, such as road works, airports and construction sites, are among the most at risk. It is recommended to take the following steps:

- assess the noise;
- take steps to prevent or control risks;
- where possible, eliminate the source of noise;
- control the effect of noise;
- provide personal protective equipment ;
- provide information and training;
- regularly monitor and analyze the effectiveness of measures.

#### **4 Economical part**

##### **4.1 Technical and economic justification for the realization of blockchain application**

The theme of the degree project focuses on the development of blockchain application for World Anti-Doping Agency (WADA), the main aim of which is the building up working product.

Modern trends, such as an increase in the number of devices connected to the Internet, exponential growth in information volumes, the development of cloud technologies are changing the telecom. There is an increase in the volumes of network traffic, and business increasingly needs to configure large-scale networks.

Now, in the absence of a miscalculation of the direct benefits from the introduction and operation of information technologies, no reasonable leader who

manages a competitive company will enter the information system into the creation without careful analysis and determining its economic efficiency and expediency. Since the cost of the error has the potential to amount to hundreds of thousands of dollars. The introduction practice has shown that the methodology and special approaches will be required to assess the economic efficiency.

This section provides an overview of the economic implementation of this work, reflecting the time, labor and financial costs of the project. In addition to technical parameters, one of the main parameters is the economic efficiency of the project. Characteristics of increasing the efficiency of net income and the discounted payback period of investments. When the project is implemented, the project must also take into account the internal rate of return and the cost of the project.

Thus, to assess the technical and economic efficiency of the project, it is necessary: to calculate the capital costs; calculate the number of employees, the cost of services; income; effective capital investments and investments.

## 4.2 Calculation of investment costs

Capital costs are calculated by the formula (5.1)

$$C = P + C_p + C_s + C_d, \quad (4.1)$$

Where:

P – price for network equipment;

$C_p$  – cost of workplaces for 1 year;

$C_s$  – cost for installation of equipment (5% of the cost of equipment);

$C_d$  – cost of the development period.

Table 4.1 – Name and value of physical equipment

Name of equipment	Quantity	Price, tenge	Cost, tenge
D-Link DGS-1100-26MPP/B1A	1	150 800	150 800
Connector RJ-45 SHIP S901E, Cat.6	40	85	3400
UTP cable 5e Cat 305m, 4-pair, D135S-P, SHIP	1	82 305	82 305
Crimping tool HT-2008 / HT-2008R (AR) / HT-200R / LY-T2008R	2	5 200	10 400
Total			246 905

We are taking into account the exchange rate as of March 25, 2018 - 320 tenge per dollar. Based on the data in Table 4.1, it can be noted that higher costs are expected, which is the reason for the urgency of training on the virtual simulator, which fully pays for these costs. [19]

Costs of equipment will be 246 905 tenge.

Table 4.2 – Calculation of costs for the organization of the workplace [20], [21], [22]

Name of equipment	Quantity	Price, tenge	Cost, tenge
Asus Zenbook UX430UQ	2	220 000	440 000
Computer desk	1	10 500	10 500
Chair	4	4 500	18 000
Table	2	14 345	28 690
Total			477 190

Total costs for the organization of workplace: 477 190 tenge.

Calculating the cost of the development of blockchain application by the formula (4.2)

$$C_d = S_{dev} + D_{PC} + P_r + C_{el} + E_n, \quad (4.2)$$

Where:

$S_{dev}$  – payment to the developer;

$D_{PC}$  – computer depreciation;

$P_r$  – monthly rent price;

$C_{el}$  – the amount for electricity during the project;

$E_n$  – cost of missing equipment and additional materials;

Electricity costs are calculated by the following formula (4.3)

$$C_{el} = W \cdot T \cdot S, \quad (4.3)$$

Where:

$W$  – consumable power  $W = 0.3$  kW;

$T$  – working hours;

$S$  – cost of 1 kW per hour of electricity  $S = 19.42$  tenge/kW-h.

On average, in one month 22 eight-hour working days,  $T = 172$  h.

Calculation of electricity costs

$$C_{el} = 0.3 \cdot 172 \cdot 19.42 = 1\,003 \text{ tenge.}$$

The power consumed for necessary needs is calculated as 5% of the power used by the main equipment. Cost of electricity for necessary needs

$$C_{el.n} = 1003 \cdot 0.05 = 50 \text{ tenge.}$$

Total energy costs

$$C_{el.total} = C_{el} + C_{el.n} = 1003 + 50 = 1\,053 \text{ tenge.}$$

The cost of additional materials is 5% of the cost of the system

$$E_n = 220\,000 \cdot 0.05 = 11\,000 \text{ tenge.}$$

Computer depreciation will be 40% of the price

$$D_{PC} = 220\,000 \cdot 0.4 = 88\,000 \text{ tenge.}$$

Calculate the cost of blockchain application according to formula (4.2)

$$C_d = 230\,000 + 88\,000 + 25\,000 + 808 + 11\,000 = 354\,808 \text{ tenge.}$$

Table 4.3 – Calculation of costs for the development

Name of expenses	Cost, tenge
Monthly salary for the developer	230 000
Computer depreciation at development time	88 000
Monthly rent of a premise in development	25 000
Energy costs during the development	1 053
Costs for additional materials and missing equipment at the time of development	11 000
Total	355 053

Calculate the capital costs by the formula (4.1)

$$C = 246\,905 + 477\,190 + 23\,750 + 355\,053 = 1\,102\,898 \text{ tenge.}$$

Table 5.4 – Capital costs for the organization of the workplace

Name of expenses	Cost, tenge	Specific weight, %
Cost of equipment	246 905	22
Cost of workplaces	477 190	43
Installation of equipment	23 750	3
Development costs	355 053	32
Total	1 102 898	100

Figure 4.1 builds the structure of capital expenditures. You can clearly see that most of the costs covers the organization of workplaces.

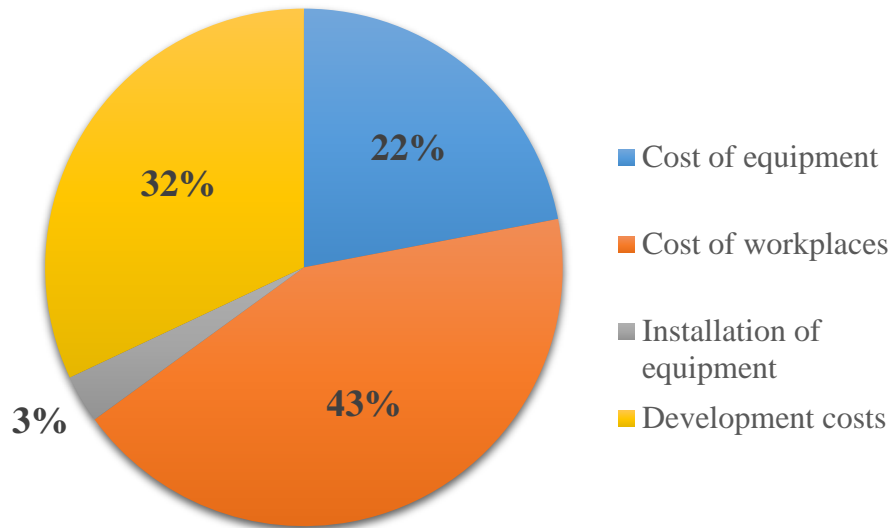


Figure 4.1 – Structure of capital costs

### 4.3 Calculation of annual operating costs

Operational costs are determined by the formula (4.4)

$$C_{op} = PF + T_s + D + C_{el} + M + C_{adm}, \quad (4.4)$$

Where:

PF – payroll fund (basic and additional wages);

$T_s$  – social tax;

D – depreciation deductions;

$C_{el}$  – electric power from the production side;

M – costs for materials and spare parts;

$C_{adm}$  – other administrative and operational expenses.

Table 4.5 - Average salary of staff per month

List of staff	Quantity	Monthly salary, tenge	Salary per year, tenge
Engineer	2	230 000	5 520 000
Total			5 520 000

Average salary of the working staff per month is given in Table 4.5 to determine the wage,

The wage fund also takes into account additional wages (payment for work on holidays, overtime, etc.) in the amount of 30% of the basic salary.

Blockchain platform is designed for a period of 1 month. Given that the operator will maintain the equipment for 9 months.

The additional wage is calculated by the formula (4.5)

$$W_a = W_p \cdot 0.3, \quad (4.5)$$

where  $W_p$  – annual basic salary fund.

Substituting these values into formula (4.5), we calculate the total sum of additional wages:

$$W_a = 5\,520\,000 \cdot 0.3 = 1\,656\,000 \text{ tenge.}$$

Payroll fund is equal to the amount of basic and additional wages:

$$PF = W_p + W_a, \quad (4.6)$$

We calculate the wage fund according to the formula (5.6)

$$PF = 5\,520\,000 + 1\,656\,000 = 7\,176\,000 \text{ tenge.}$$

Deductions for social tax are from 9.5%:

$$T_s = 0.095 \cdot (PF - 0.1 \cdot PF) = 0.095 \cdot 6\,458\,400 = 613\,548 \text{ tenge.}$$

The amount of depreciation is accounted according to the established norms, which are calculated as a percentage of the value of fixed assets (4.7)

$$D_0 = \frac{F \cdot N_D}{100\%}, \quad (4.7)$$

Where:

$F$  – carrying value of fixed assets, tenge;

$N_D$  – depreciation rate;

Calculate the depreciation of equipment and office furniture using the formula (4.7).

Depreciation of computer equipment is 40% of the price:

$$D_1 = 440\,000 \cdot 0.4 = 176\,000 \text{ tenge.}$$

Depreciation of office furniture is 15% of the price:

$$D_2 = 57\,190 \cdot 0.15 = 8\,579 \text{ tenge.}$$

$$D = D_1 + D_2 = 176\,000 + 8\,579 = 184\,579 \text{ tenge.}$$

Cost of electric power is calculated by the formula (4.8)

$$C_{el} = W \cdot T \cdot S, \quad (4.8)$$

Where:

$W$  – consumable power  $W = 1 \text{ kW}$ ;

$T$  – hours of work for 1 year,  $T = 1\,968 \text{ hours per year}$ ;

$S$  – cost of 1 kW per hour of electricity  $S = 19.43 \text{ tenge/kW-h.}$

Define the cost of electricity:

$$C_{el} = 1 \cdot 1\,968 \cdot 19.43 = 38\,238 \text{ tenge.}$$

The power consumed for necessary needs, for example light, or occasionally the inclusion of an air conditioner is calculated as 10% of the power used by the main equipment. The cost of electricity for the necessary needs:

$$C_{el.n} = C_{el} \cdot 0.1 = 38\,238 \cdot 0.1 = 3\,823 \text{ tenge.}$$

Total energy costs:

$$C_{el.total} = C_{el} + C_{el.n} = 38\,238 + 3\,823 = 42\,061 \text{ tenge.}$$

The cost of additional materials and missing equipment is calculated as 5% of the cost of the system:

$$M = 477\,190 \cdot 0.05 = 23\,860 \text{ tenge.}$$

Expenses of additional costs is 10% of the operational costs:

$$C_{adm} = 477\,190 \cdot 0.1 = 47\,719 \text{ tenge.}$$

Consequently, the operating costs based on the formula (4.4) will be:

$$C_{op} = 7\,176\,000 + 613\,548 + 184\,576 + 23\,860 + 42\,061 + 47\,719 = 8\,087\,764 \text{ tenge.}$$

The most part of the operating costs is occupied by the payroll fund. Important in these costs are depreciation charges. The smallest part of the operating costs is allocated to electricity because of the selection of energy-saving equipment.

Enter the data on operating costs in Table 4.5 and calculate the specific weight of each cost.

Table 4.6 – Operating costs

Operating cost Items	Cost, tenge	Specific weight, %
Payroll fund	7 176 000	88
Social tax	613 548	7.5
Depreciation deductions	184 576	3
Costs for materials and spare parts	23 860	0.4
Electricity costs	42 061	0.5
Other expenses	47 719	0.6
Total	8 087 764	100

Figure 4.2 represents the structure of operating costs.

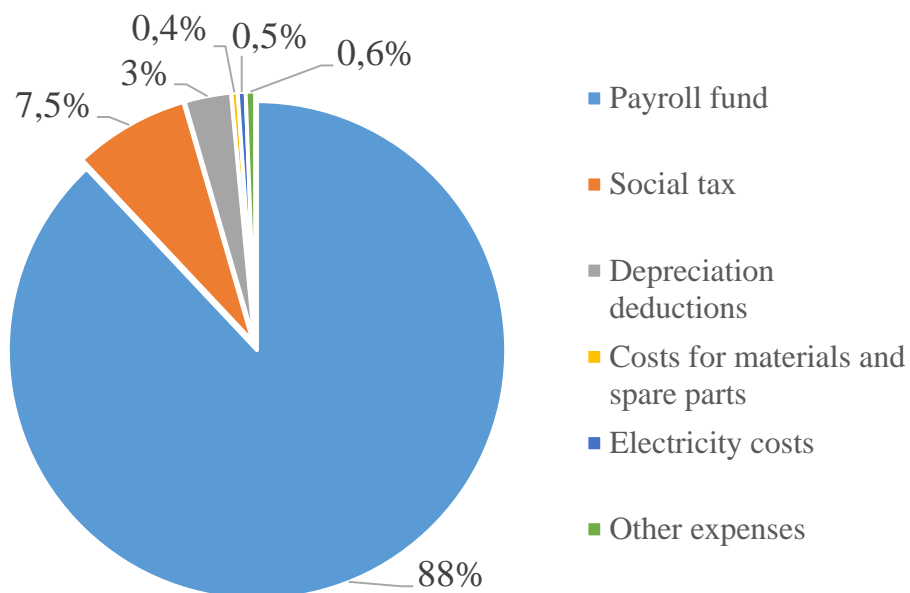


Figure 4.2 – Structure diagram of operating costs

#### 4.4 Calculation of income

It is calculated that the courses will be 60 people, given that in the academic year is 12 months, the cost of tuition is expected to be set at 20 000 tenge per person.

$$I = 20\,000 \cdot 60 \cdot 12 = 14\,400\,000 \text{ tenge.}$$

As a result, we get that the income will be 14.4 million tenge.

#### 4.5 Calculation of economic efficiency

The profit from the introduction of these rates is the income from the core business, net of operating expenses. Net profit is determined by deducting income tax of 30% (for the Republic of Kazakhstan)

We calculate the profit of the enterprise before taxation.

Profit from the introduction of training courses is calculated by formula (4.9)

$$P = I - C_{op}, \quad (4.9)$$

Where:

$I$  – annual income;

$C_{op}$  – operating costs

$$P = 14\,400\,000 - 8\,087\,764 = 6\,312\,236 \text{ tenge.}$$

The net income remaining available is profit taking into account income tax.

The calculation of corporate tax will be 20% of the profit under the formula (4.10)

$$CT = P \cdot 20\%, \quad (4.10)$$

$$CT = 6\,312\,236 \cdot 0.2 = 1\,262\,447 \text{ tenge.}$$

The amount of net profit taking into account income tax according to the formula (4.11) will be

$$NP = P - CT, \quad (4.11)$$

$$NP = 6\,312\,236 - 1\,262\,447 = 5\,049\,789 \text{ tenge.}$$

Absolute economic efficiency will be calculated using formula (4.12)

$$E = NP/C_{\Sigma}, \quad (4.12)$$

$$E = 5\,049\,789 / 1\,102\,898 = 4.57$$

The payback period is the reciprocal of the absolute economic efficiency and is given by formula (4.13)

$$T = 1/E, \quad (4.13)$$

$$T = 1/4.57 = 0.21 \text{ year.}$$

Based on the received data, we get that the costs for this project will pay off in approximately 3 months.

All the economic indicators for the project are shown in Table 4.7

Table 4.7 – Indicators of economic efficiency of the project

Indicator	Value
Capital costs, tenge	1 102 898
Operating costs, tenge	8 087 764
Profit before taxation, tenge	6 312 236
Profit after taxation, tenge	5 049 789
Economic efficiency	4.57
Payback period, months	3 months

Taking into account all the data obtained, in the development of blockchain application for World Anti-Doping Agency (WADA) with capital expenditures in the amount of 1 102 898 tenge, net annual income will amount to 5 049 789 tenge. This project pays off for 3 months. Consequently, it can be concluded that the development of this application is cost-effective.

## **Conclusion**

This final qualifying work was devoted to the analysis of the principles of construction, algorithms and data structures of the blocking technology.

The blockchain is a new way of technological advancement. It can achieve secure transactions without the need for a central management agency. Since 2009, Bitcoin has used blockchain technology. Most importantly, the new application is based on the basic elements of Blockchain technology rather than the currency area. The first application of digital mine distribution, including all transactions, is a digital currency.

The main aspect of this work was the development of an application for WADA based on the Hyperledger Sawtooth framework. This framework allows you to create, store, transfer data decentralized with the inability to further edit. Thus, showing the whole essence and relevance of creating a block of applications.

Also in the work were considered the structure of the block with all its components: hash functions, blocks, transactions, encryption keys, hard forks and soft forks. The process of installing the Hyperledger Sawtooth framework was demonstrated with the help of which this application was developed with all the components making it possible to interact without interference. In addition to creating the application, calculations were made for the reliability of the system and the sizes of the blocks were calculated, depending on their size.

In the section of safety and vital activity, a plan of the premises was designed in which all the computational and design work was carried out to create the blocking application. Artificial and natural light, electrical safety and fire safety have formed the basis of calculations.

In the section of the economy there were calculations of the costs for the purchase and installation of equipment, staff salaries, economic efficiency and expediency of the drafting of the application and further maintenance.

## **List of abbreviations**

1. WADA – World Anti-Doping Agency
2. HS – Hyperledger Sawtooth
3. HF – Hyperledger Fabric
4. HB – Hyperledger Burrow
5. HI – Hyperledger Iroha
6. P2P – Peer-to-peer
7. PoET – Proof of Elapsed Time
8. PoW – Proof of Work
9. PoS – Proof of State
10. SBFT – Simplified Byzantine Fault Tolerant
11. YAC – Yet another Consensus
12. REST – Representational State Transfer
13. CLI – Command Line Interface
14. FTP – File Transport Protocol
15. HTTP – HyperText Transfer Protocol
16. JS – JavaScript
17. IoT – Internet of Things
18. SDK – Software Development Kit
19. HTML – HyperText Markup Language
20. JSON – JavaScript Object Notation
21. MTBF - Mean Time Between Failure
22. MTTR – Mean Time To Repair
23. API – Application Programming Interface
24. IP – Internet Protocol
25. RAM – Random Access Memory
26. CPU – Central Processing Unit
27. TB – Terabyte
28. GB – Gigabyte
29. UI – User Interface
30. OSI – Open System Interconnection
31. URL – Uniform Resource Locator
32. DLT – Distributed Ledger Technology
33. DB – DataBase
34. KYC – Know Your Customer
35. DAO – Decentralized Autonomous Organization

## List of references

1. Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S., Bitcoin and
2. Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
3. Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
4. <https://bitcoin.org/bitcoin.pdf>
5. Polyzos G. C., Fotiou N. Blockchain-assisted Information Distribution for the Internet of Things //2017 IEEE International Conference on Information Reuse and Integration (IRI). – IEEE, 2017. – С. 75-78.
6. <https://www.hyperledger.org/category/hyperledger-sawtooth> (date of request 25.03.2018)
7. Bakre A., Patil N., Gupta S. Implementing Decentralized Digital Identity using Blockchain. – 2017.
8. Zheng Z. et al. An overview of blockchain technology: Architecture, consensus, and future trends //Big Data (BigData Congress), 2017 IEEE International Congress on. – IEEE, 2017. – С. 557-564.
9. Wong, J. and Kar, I., "Everything you need to know about the Ethereum 'hardfork,'" Quartz Media, July 18, 2016. <https://qz.com/730004/everything-you-need-to-know-about-the-ethereum-hard-fork/>
10. Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., and Smith-Tone, D., National Institute of Standards and Technology (NIST), NIST Internal Report (NISTIR) 8105, Report on Post-Quantum Cryptography, April 2016. <https://doi.org/10.6028/NIST.IR.8105> (date of request 23.04.2018)
11. <https://github.com/hyperledger/sawtooth-core> (date of request 05.05.2018)
12. Mell, P., Kelsey, J., and Shook, J., "Cryptocurrency Smart Contracts for Distributed Consensus of Public Randomness." October 7, 2017. [https://doi.org/10.1007/978-3-319-69084-1\\_31](https://doi.org/10.1007/978-3-319-69084-1_31)
13. National Institute of Standards and Technology (NIST), report on blockchain technology review – January 2018
14. СНиП РК 2.04-05-2002 «Естественное и искусственное освещение. Общие требования»
15. Абдимуратов Ж.С., Манабаева С.Е. Безопасность
16. жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009.
17. Маринченко А.В. Безопасность жизнедеятельности: Учебное пособие. – 2-е изд., доп. и перераб. – М.: Издательско-торговая корпорация «Дашков и К», 2007.

18. Т. Е. Хакимжанов. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. - Алматы: АИЭС, 2002.
19. Базылов К.Б., Алибаева С.А., Бабич А.А. Методические указания по выполнению экономического раздела дипломной работы бакалавров для студентов всех форм обучения специальности 050719 – Радиотехника, электроника и телекоммуникации. – Алматы: АИЭС, -2009. -19 с.
20. Голубицкая Е.А. Экономика связи: М.: -Ирмас, 2006.
21. Фурсов В.Г. Финансовый менеджмент: конспект базовых лекций. - М.: МАИ, 2010.-125с.
22. Alekseeva N. A. “Deployment and improvement of reproductive chains as a result of the influence of the intellectual capital of small business // Multi-level social reproduction: questions of theory and practice” (2010) P. 7-11.
23. Tumanov D V “Development of the Information Society, Role in the Reproductive Process. // Multilevel Public Reproduction: Questions of Theory and Practice”( 2013) P. 291 -300.
24. <https://github.com/hyperledger/sawtooth-supply-chain> (date of request 01.04.2018)
25. Vakhrushev D. S. “Self-organization and dynamic stability of economic systems: theoretical and methodological aspects” (2009) P. 345-407
26. Badev, A and M Chen : “Bitcoin: technical background and data analysis”, Finance and Economics Discussion Series, (2014) p.104
27. Chiu, J and T-N Wong: “E-money: efficiency, stability and optimal policy”, (2014) p. 12-14
28. Fung, Band H Halaburda : “Understanding platform-based crypto currencies”, (2014) p. 12–20
29. Gandal, N and H Halaburda : “Competition in the cryptocurrency market” (2014), pp. 33
30. Ciaian, P., Rajcaniova, M., & Kancs, D. A. The economics of Bitcoin price formation. Applied Economics. (2016): p.1799-1815.
31. Corbet S, Lucey B, Yarovya L. Datestamping the Bitcoin and Ethereum bubbles. Finance Research Letters. (2017) p.433-437.
32. Aggelos Kiayias, Elias Koutsoupas, Maria Kyropoulou, and Yiannis Tselekounis. Blockchain mining games. In Proceedings of the 2016 ACM Conference on Economics and Computation, pages 365–382. ACM, 2016.
33. Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In International Conference on Financial Cryptography and Data Security, pages 264–279. Spring, 2017.
34. D. Monderer and L.S. Shapley. Potential games. Games and Economic Behavior, 14:p.124–143,1996.
35. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, p 6-8, 2008.

## Appendix A

### Listing of app.js program

---

```
// Select User
$('[name="keySelect"]').on('change', function () {
  if (this.value === 'new') {
    app.user = makeKeyPair()
    app.keys.push(app.user)
    saveKeys(app.keys)
    addOption(this, app.user.public, true)
    addOption('[name="transferSelect"]', app.user.public)
  } else if (this.value === 'none') {
    app.user = null
  } else {
    app.user = app.keys.find(key => key.public === this.value)
    app.refresh()
  }
})

// Create Asset
$('#createSubmit').on('click', function () {
  var asset = $('#createName').val()
  asset = asset.concat(" ", $('#boxNo').val() " ", $('#missionOrder').val() " ", $('#timeSealed').val() " ");
  console.log(asset);
  const boxNo = $('#boxNo').val()
  const missionOrder = $('#missionOrder').val()
  const timeSealed = $('#timeSealed').val()
  if (asset) app.update('create', asset)
})

// Transfer Asset
$('#transferSubmit').on('click', function () {
  const asset = $('[name="assetSelect"]').val()
  const owner = $('[name="transferSelect"]').val()
  if (asset && owner) app.update('transfer', asset, owner)
})

// Accept Asset
$('#transferList').on('click', '.accept', function () {
  const asset = $(this).prev().text()
  if (asset) app.update('accept', asset)
})

$('#transferList').on('click', '.reject', function () {
  const asset = $(this).prev().prev().text()
  if (asset) app.update('reject', asset)
})
```

## Appendix B

### Listing of handlers.js program

```
const { createHash } = require('crypto')
const { TransactionHandler } = require('sawtooth-sdk/processor')
const { InvalidTransaction } = require('sawtooth-sdk/processor/exceptions')
const { TransactionHeader } = require('sawtooth-sdk/protobuf')

// Encoding helpers and constants
const getAddress = (key, length = 64) => {
  return createHash('sha512').update(key).digest('hex').slice(0, length)
}

const FAMILY = 'transfer-chain'
const PREFIX = getAddress(FAMILY, 6)

const getAssetAddress = name => PREFIX + '00' + getAddress(name, 62)
const getTransferAddress = asset => PREFIX + '01' + getAddress(asset, 62)

const encode = obj => Buffer.from(JSON.stringify(obj, Object.keys(obj).sort()))
const decode = buf => JSON.parse(buf.toString())

// Add a new asset to state
const createAsset = (asset, owner, state) => {
  const address = getAssetAddress(asset)

  return state.get([address])
    .then(entries => {
      const entry = entries[address]
      if (entry && entry.length > 0) {
        throw new InvalidTransaction('Asset name in use')
      }

      return state.set({
        [address]: encode({name: asset, owner})
      })
    })
}

// Add a new transfer to state
const transferAsset = (asset, owner, signer, state) => {
  const address = getTransferAddress(asset)
  const assetAddress = getAssetAddress(asset)

  return state.get([assetAddress])
    .then(entries => {
      const entry = entries[assetAddress]
      if (!entry || entry.length === 0) {
        throw new InvalidTransaction('Asset does not exist')
      }
    })
}
```

## Appendix C

### Listing of index.html file

```

<header>
  <div id="left_header">ADAMS</div>
  <i id="right_header">World Anti-Doping Agency</i>
</header>

<div id="body">
  <div id="left-column">
    <div class="form-group">
      <label>Doping Control Officer's public key</label>
      <select class="form-control" name="keySelect">
        <option value="none" selected>Select public key...</option>
        <option value="new">Create new DCF</option>
      </select>
    </div>

    <div class="form-group">
      <label>Doping Control Form's components</label>
      <input id="createName" class="form-control" type="text" placeholder="Enter athlete name...">
      <input id="boxNo" class="form-control" type="text" placeholder="Enter box number...">
      <input id="missionOrder" class="form-control" type="text" placeholder="Enter mission order...">
      <input id="timeSealed" class="form-control" type="text" placeholder="Enter sealed time...">
      <input id="createSubmit" type="button" value="Create" class="btn btn-primary">
    </div>

    <div class="form-group">
      <!--
      <label>Transfer Doping Control Form</label>
      <select class="form-control" name="assetSelect">
        <option value="none" selected>Select Doping Control Form...</option>
      </select>
      <select class="form-control" name="transferSelect">
        <option value="none" selected>Select ADAMS/DCO recipient...</option>
      </select>
      <input id="transferSubmit" type="button" value="Transfer" class="btn btn-primary">
    </div>

    <div class="form-group">
      <label>Accept DCF</label>
      <div id="transferList"></div>
    </div>
  </div>
  ->|
  <div id="data">
    <label>ADAMS List</label>
    <table class="table table-hover">
      <tr>
        <th>Athlete name</th>
        <th>Box number</th>
        <th>Mission order</th>
      </tr>
    </table>
  </div>

```