

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: Designing of a platform for distributed register

Specialty: 5B071900 – Radio engineering electronics and telecommunications

Implemented by: Sosnin K.A. ICT-14-9
(Student's surname and initials) group

Scientific Supervisor: Panchenko S.V., M.S., senior lecturer
(Surname and initials, degree, rank)
«30» 05 2018 y.
(sign)

Advisors:
of Economy section: Tuzelbayev B.I., PhD, associate professor
(Surname and initials, degree, rank)
«31» 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)
«30» 05 2018 y.
(sign)

Standards compliance controller: Panchenko S.V., M.S., senior lecturer
(Surname and initials, degree, rank)
«30» 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: Sosnin Kirill Andreevich
(name, patronymic and surname)

Theme: Designing of a platform for distributed register

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 diploma project are fast development of distributed ledger technologies. Having gained a great reputation in the field of cryptocurrency, the blockchain develops rapidly. In this diploma project, it is proposed to create an application for keeping records of students' academic performance that will work on the basis of distributed register technology.

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

In this diploma project, we studied the basic principles of blockchain technology. In the practical implementation section, the Hyperledger Fabric platform was chosen and its advantages were explored. The implemented application fully reflects the main principles of distributed ledger technology. Also, main parameters of the constructed network were calculated, such as the time of one transaction and fault tolerance.

In the life safety activity section, optimal working conditions were calculated.

In the economic part of the project, economic efficiency and costs for implementation of this project were calculated.

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

Blockchain block data structure

Block hashing

Structure of Smart contracts

The roles of participants in the Hyperledger Fabric network

Illustration of one possible transaction flow

Architecture of developed application

Creating a record in the client part

Hyperledger Fabric docker images

Hyperledger Fabric sample network

User interface of application

Recommended main references:

Ali M. Trust-to-trust design of a new Internet, 2017.

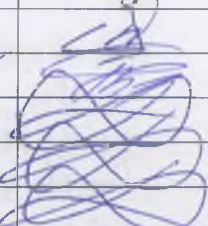
Iansiti M, Lakhani K.R. The truth about blockchain, Harvard Business Review, 2017.

Hyperledger-fabric docs Documentation, Linux Foundation, 2015

Shanti B. Blockchain, research paper, 2017.

Hackius N, Petersen M. Blockchain in logistics and supply chain: trick or treat, 2017

Project adviser with corresponding sections specifying:

Section	Advisor	Dates	Sign
Life activity safety	Beginbetora A.S	10.03 - 25.05.18	
Economy section	Tuzelbayev B.T	30.04 - 31.05.18	
Computer science	Panchenko S.V.	01.02 - 30.05.18	
Technical part	Panchenko S.V.	12.03 - 30.05.18	
Standard Compliance	Panchenko S.V.	02.05 - 30.05.18	

SCHEDULE of diploma project implementation

№	Sections, list of developing questions	Dates of bringing to Scientific Supervisor	Notes
1	Introduction	25.12.17 - 04.01.18	Done
2	Distributed ledger technology overview	05.01.18 - 10.01.18	Done
3	Characteristics of blockchain	11.01.18 - 19.01.18	Done
4	Hyperledger technologies overview	20.01.18 - 31.01.18	Done
5	Formulation of the problem	01.02.18 - 03.02.18	Done
6	Network structure and architecture of application	04.02.18 - 15.02.18	Done
7	Creating an application	16.02.18 - 05.03.18	Done
8	Making a changes into a client part	06.03.18 - 10.03.18	Done
9	Calculation of main network parameters	11.03.18 - 16.03.18	Done
10	Further development planning	16.03.18 - 17.03.18	Done
11	General information on the labour protection	19.03.18 - 23.03.18	Done
12	Calculation of a microclimate	24.03.18 - 29.03.18	Done
13	Fire safety	29.03.18 - 31.03.18	Done
14	Conclusion of life safety part	01.05.18 - 05.05.18	Done
15	Calculation of investment costs	02.04.18 - 10.04.18	Done
16	Calculation of income	11.04.18 - 15.04.18	Done
17	Calculation of economic efficiency	22.04.18 - 30.04.18	Done
18	Conclusion	11.05.18 - 23.05.18	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: _____

Sosnin K. A

(Surname and initials)

Андатпа

Бұл дипломдық жобада blockchain технологиясы негізінде жасалған студенттердің үлгерімін есептейтін, классикалық клиент-сервер қисынды деректерді сақтау тәсілінен ерекше қолданбаны жасау қаралады. Жасалатын қолданбаның негізінде Hyperledger Fabric таңдалды. Қойылған мақсаттардың тәжірибелік бөлімі ретінде қолданбаның клиенттік бөлшегі, сонымен қатар кіріс транзакциялардың түсу және өңдеу логикасы бейімделген. Жобаның техникалық есептері, және оңтайлы еңбек жағдайлары есептелген. Жобаның экономикалық бөлімінде қолданбаның экономикалық енгізу тиімділігі есептеледі.

Аннотация

В данном дипломном проекте рассматривается создание приложения по учету успеваемости студентов на основе технологии blockchain, принципиально отличающегося от классических решений с клиент-серверной логикой методами хранения данных. В качестве базы для создаваемого приложения выбран Hyperledger Fabric. Для практической реализации поставленных задач были адаптированы клиентская часть приложения, а также логика поступления и обработки входящих транзакций. Были произведены технические расчеты проекта, а также расчет оптимальных условий труда. В экономической части проекта рассчитывается экономическая эффективность внедрения приложения.

Abstract

In this diploma project, it was considered the creation of an application for recording students' progress on the basis of blockchain technology, which fundamentally differs from classical solutions with client-server logic by data storage methods. As the base for the application being created was selected Hyperledger Fabric. For practical implementation of the tasks, was adapted the client part of the application, as well as the logic of incoming and processing of transactions. Technical calculations of the project were carried out, as well as calculation of optimal working conditions. In the economic part of the project, the economic efficiency of introducing the application is calculated.

Content

Introduction	7
1 Theoretical part	8
1.1 Distributed Ledger Technology	8
1.2 Characteristics of blockchain	10
1.3 Smart Contracts	16
1.4 IOTA	19
1.5 Ethereum Smart Contracts	20
1.6 Hyperledger	21
2 Practical realization of the project	23
2.1 Formulation of the problem	23
2.2 Justification of the choice of Hyperledger Fabric	24
2.3 Transaction flow	27
2.4 Network structure and architecture of application	32
2.5 Process of creating an application for the recording of students' progress	36
2.6 Calculation of main network parameters	43
2.7 Further development	48
3 Life activity safety section	48
3.1 General information on the labor protection of the enterprise	48
3.2 Measuring of noise and vibration level	50
3.3 Calculation of microclimate parameters	52
3.4 Fire safety	55
3.5 Calculation of ventilation	58
3.6 Conclusion of the section	59
4 Economical part	60
4.1 Technical and economic justification for the designing of the platform for distributed ledger application	60
4.2 Calculation of investment costs	61
4.3 Calculation of annual operating costs	63
4.4 Calculation of income	66
4.5 Calculation of economic efficiency	66
Conclusion	69
List of abbreviations	70
List of references	71
Appendix A Listing of the startFabric.sh program	73
Appendix B Listing of the registerUser.js program	74
Appendix C Listing of the server.js program	75
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

Looking back to the last half century of computer technologies and architectures, one may observe a trend of fluctuation between the centralization and subsequent decentralization of computing power, storage, infrastructure, protocols, and code

The mainframe was the main part of computer network. They usually retain all the computing power, memory, data storage, and code. Access to the mainframe is primarily through the 'dumb terminal', which requires only input and output, and cannot store or process data [1].

With the advent of personal computers and private networks, customers now have the same computing power and servers are placed in their own computer. In summary, the "client-server" architect broke out and has supported the development of the decentralized database system. Large datasets placed in mainframes can go to distributed architectures. This data could replicate from one server to another, and subsets of the data could be situated and running on clients, and then, sent back to the server.

Over time, the Internet and cloud computing architectures support global access to a variety of computing devices; mainframes are designed primarily to meet the needs of large companies and governments. Although the "hardware" in the cloud architecture, the level of concentration of applications is decentralized (for example, Facebook, Google, etc.).

Today's transaction network is slightly more than the latest version of the network since the business records have been stored. Members of the network interact with each other, but keep separate records of their transactions. What is blockchain differences, is made the corporate network identity, implementation trading, and standard methods to store data? The establishment of the source of the property, once it is written, can be determined by looking at the list of transactions, changes are not possible. Each participant account has its own copy of the current state of network.

The purpose of this diploma project is the study of blockchain technology, the technical aspects of this technology, its practical implementation and its further use. For the implementation, it was chosen to create an application for recording students' progress. To achieve this purpose, it is necessary to solve the following tasks:

- study the technical fundamentals of technology;
- familiarity with existing platforms offering blockchain solutions;
- choice of platform for implementation of a project;
- creating an application for recording students' progress based on chosen platform;
- calculate the economical parameters and life activity safety parameters.

1 Theoretical part

1.1 Distributed Ledger Technology

Decentralized networks with peers are not new. P2P networks, like other distributed systems, have to solve a very complex informatics problem: conflict resolution, or reconciliation. Relational databases offer referential integrity, but there is no such feature in the distributed system. If two incompatible facts arrive at the same time, the system must have rules for determining which fact is correct.

There are basically two types of network architectures:

1. Client-server network;
2. Peer to peer network.

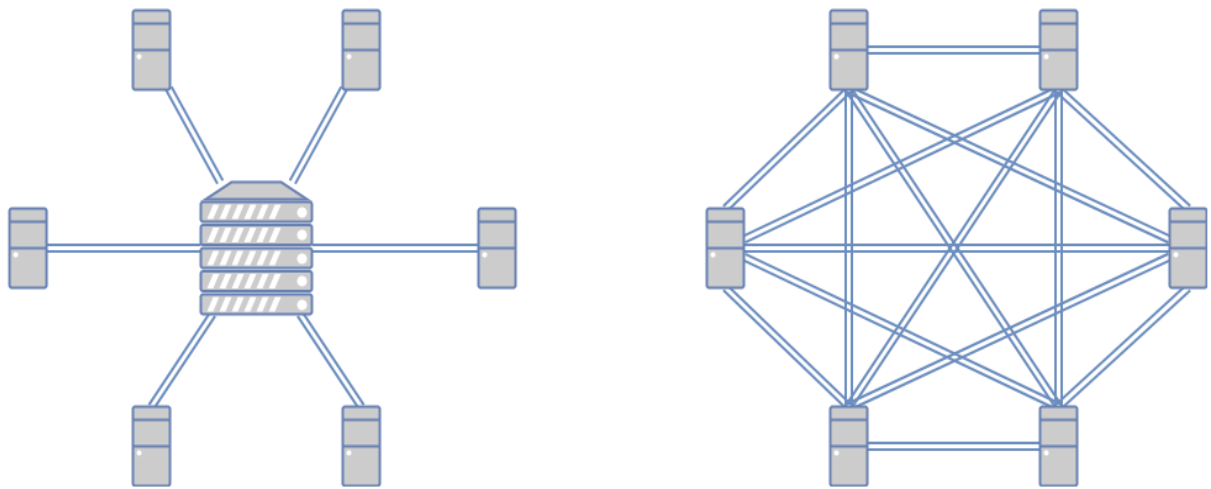


Figure. 1.1 - Client-server network and P2P network.

Networking in the first way implies centralized control of everything: services, access, data. All system logic and information are hidden inside the server, which allows you to reduce the performance requirements for client devices and ensure high processing speed. It is this method that has become most widespread today.

Peer-to-peer or decentralized networks do not have a main device, and all participants have equal rights. In this model, each user is not only a consumer, but he himself becomes a service provider

The vast majority of applications and systems for normal operation require the ability to operate a set of data. There are a lot of ways to organize such work, and one of them uses the peer-to-peer network method [2]. Distributed, or parallel, databases differ in that information in a partial or full composition is stored on each device network.

One of the advantages of this system is the availability of data: there is no single point of failure point, as in the case of a database located on one server. Such a decision also entails certain restrictions on the speed of data update and distribution among the network participants. Such a system will not stand the load of millions of users who constantly publish new information.

A distributed ledger is a type of data structure which resides across multiple computer devices, generally spread across locations or regions.

Distributed Ledger Technology includes not just blockchain and smart contracts. Distributed ledgers existed before Bitcoin, the Bitcoin blockchain represents the convergence of a multitude of technologies, which includes timestamping of transactions, Peer-to-Peer (P2P) networks, hash features, shared computational strength, together with a brand new consensus algorithm.

Blockchain is actually among the kinds of distributed information storage, uses three previously recognized technologies: peer-to-peer networks, databases along with encryption. The database is actually a chain of blocks, that is encrypted in a specific way and saved on almost all nodes of the network in the exact same type (replication is actually an actual copy). The entire secret is based on the links between the blocks as a result of cryptography, as a consequence it's nearly impossible to forge info in the blocks. Blockchain lets you easily distribute and or maybe method information between several individuals via an untrusted networking. Information could be something, but by far the most fascinating choice is the capability to transmit info that will require a third reliable party. Examples of that info are actually cash (require bank participation), property rights (require notary participation), loan agreement, etc. Essentially, the blockage removes the demand for the involvement of a third trustee.

Blocking know-how consists of the usage of a distributed data source of blocks, which are actually a linked checklist (each next block has the identifier of the prior one). Each and every part of the network will keep a copy of all operations done for all time. This will be impossible with no particular innovative developments created to make certain the safety as well as effectiveness of the network[3].

Regardless of the reality that the curiosity in blokchan technology is much more associated with the area of finance, the scope of distributed registry engineering isn't restricted to it. Together with banks as well as finteh start ups, players from various other non financial marketplaces have turned the attention of theirs to technology and are searching for solutions to make the most of the possibilities that it offers. Let us go over several exciting examples of practical uses of blocking technology which exist outside the scope of financial services.

New projects on the blockchain is going to be based on its primary benefits - openness, security, security.

Therefore, the blocking system will be a great help for any services where users could worry about possible fraud or data security:

- micropayments;
- bank operations;
- logistics;
- jurisprudence;
- medicine.

In the telecommunications industry, the block can be used in the following areas: digital identity, data management (storage of various documents, insurance, air tickets, etc.), roaming (exclude roaming-fraud), 5G (selection of the fastest

node for communication), Smart City useful in view of openness and transparency), mobile trading, M2M and IoT (secure P2P connection for IoT devices to create an economically viable and self-managing system), eHealth (secure storage of medical information in electronic form). The role of the blockade in the security of telecommunications networks boils down to the fact that distributed registry technology safely and quickly stores the exchange data and the results of cooperative device activity in the system of blockades. In the case of hacking the network, this does not affect the functioning of the entire network. Blocking in telecommunications will be the tool that will help telecom operators to transform themselves into a true Digital Service Provider. Blockchain helps to build ecosystems based on partnership interaction. Block technology allows CSP to find a profitable position in this ecosystem. The technology of blockchain is only at the beginning of its evolution. The next 5 years will be conducted experiments, verification and standardization of the block. But telecom companies need to concentrate on the application and implementation of this technology now.

This information about the blockade helps to look at the world of computer technologies, measure their internal development potential, make another step towards the evolution of thinking.

To sum up, distributed ledger technologies typically consists of 3 primary components:

- an information design which captures the present state of the ledger;
- a dialect of transactions which changes the ledger state;
- a process used to develop consensus among participants about that transactions will be acknowledged, and also in what purchase, by the ledger.

1.2 Characteristics of blockchain

1.2.1 Structure of block. Blockchain is a specific form or subset of distributed ledger technologies, which constructs a chronological chain of blocks, hence the name 'block-chain'. A block refers to a set of transactions that are bundled together and added to the chain at the same time. In the Bitcoin blockchain, the miner nodes bundle unconfirmed and valid transactions into a block. Each block contains a given number of transactions. In the Bitcoin network, miners must solve a cryptographic challenge to propose the next block. This process is known as 'proof of work', and requires significant computing power [4].

Each block consists of an address, the date and time of creation, a hash, and a list of transactions:

- address - a public key generated by an asymmetric encryption algorithm (for example, RSA), based on a private key invented by the user;
- date and time - the moment when the block was created (the transaction also has a date and time of creation);
- hash - is calculated from the address of the previous block and the hash sum of all transactions of the current block, why the binder? Because it calculates the hash of the previous block;

- information - a message, the amount of money (crypto-currencies), documents, the history of diseases, program code (smart contracts), etc.

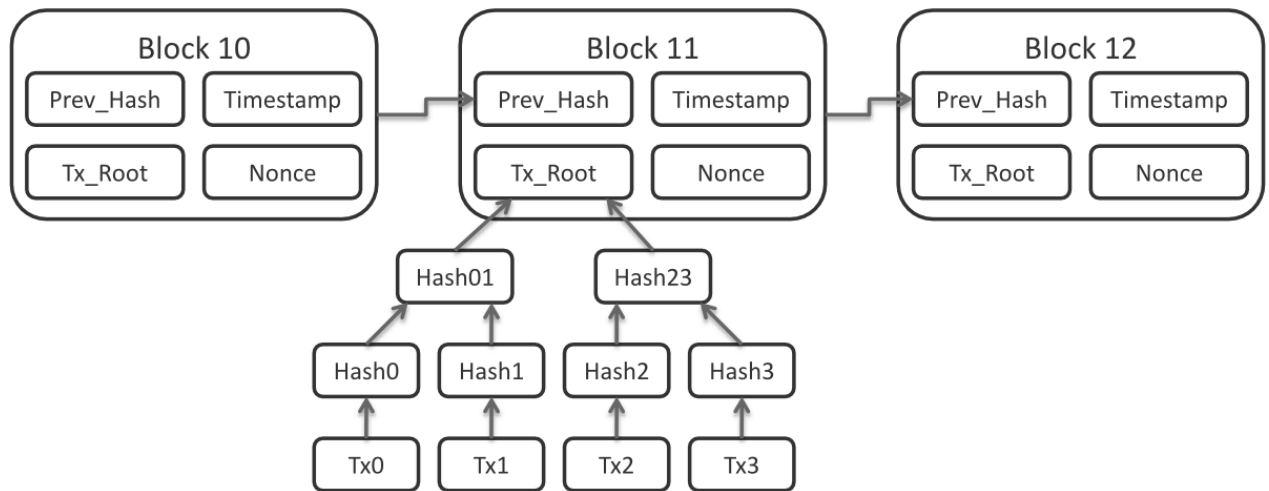


Figure 1.2 - Bitcoin Block Data Structure

The record of an event, cryptographically secured with a digital signature, that's verified, ordered, and bundled together into blocks, form the transactions in the blockchain. In the Bitcoin blockchain, transactions involve the transfer of bitcoins, while in other blockchains, transactions may involve the transfer of a record or perhaps any asset of some service being rendered [6]. Moreover, a smart contract within the blockchain may allow automatic execution of transactions upon meeting predefined criteria.

1.2.3 Cryptography. Cryptography has a crucial role to play both in the security, as well as in the immutability of the transactions recorded on blockchains. Cryptography is definitely the study of the methods used in order to allow secure communication between many different parties as well as to ensure the authenticity and immutability of the data being communicated. For blockchain technologies, cryptography is actually used to confirm that a transaction was developed by the right person. It's also used to link transactions into a block in a tamper proof way, as well as create the links between blocks, to form a blockchain.

Cryptography is the heart of the detachment, which ensures the operation of the system. The architecture of the blockade assumes that the trust between the participants of the network is based on the principles of mathematics and economics, that is, it is formalized. Cryptography also guarantees security, which is based on the transparency and verifiability of all operations, rather than on the industry's traditional perimeter security system (perimeter security) [8].

Various cryptographic techniques guarantee the invariability of the transaction log of the block, resolve the authentication task and control the access to the network and data in the blockchain as a whole. In today's material, we'll talk about hash functions, keys and digital signatures. The immutability on the information and that rests on the blockchain is probably one of the most potent as well as persuading motive to deploy blockchain based ways for a wide range of socio economic tasks that are presently captured on centralized servers. This

particular immutability, or 'unchanging more than time' feature of the brother can make the blockchain helpful for accounting, monetary transactions, identity managing, as well as advantage ownership, transfer and management, simply to name a couple of cases. When a transaction is actually created upon the blockchain, nobody is able to alter it, or perhaps, at any rate, it will be incredibly hard to alter it.

It's incredibly difficult to modify the transactions inside a blockchain, since every obstruct is actually associated with the prior obstruct by like the previous block's hash. This hash consists of the Merkle root hash of most of the transactions within the prior obstruct. In case one transaction had been changing, not merely would the Merkle root hash alter, but therefore as well would the hash found in the evolved obstruct. Additionally, every consequent obstruct will have to become kept up to date to mirror the shift. Within the situation of evidence of perform, the quantity of electricity needed to recalculate the nonce because of this obstruct and also every consequent obstruct will be prohibitive [5]. On the flip side, in case another person did change a transaction inside an obstruct without any analyzing the needed measures to upgrade the consequent blocks, it will be simple to recalculate the hashes applied to the blocks and find out which food is actually amiss.

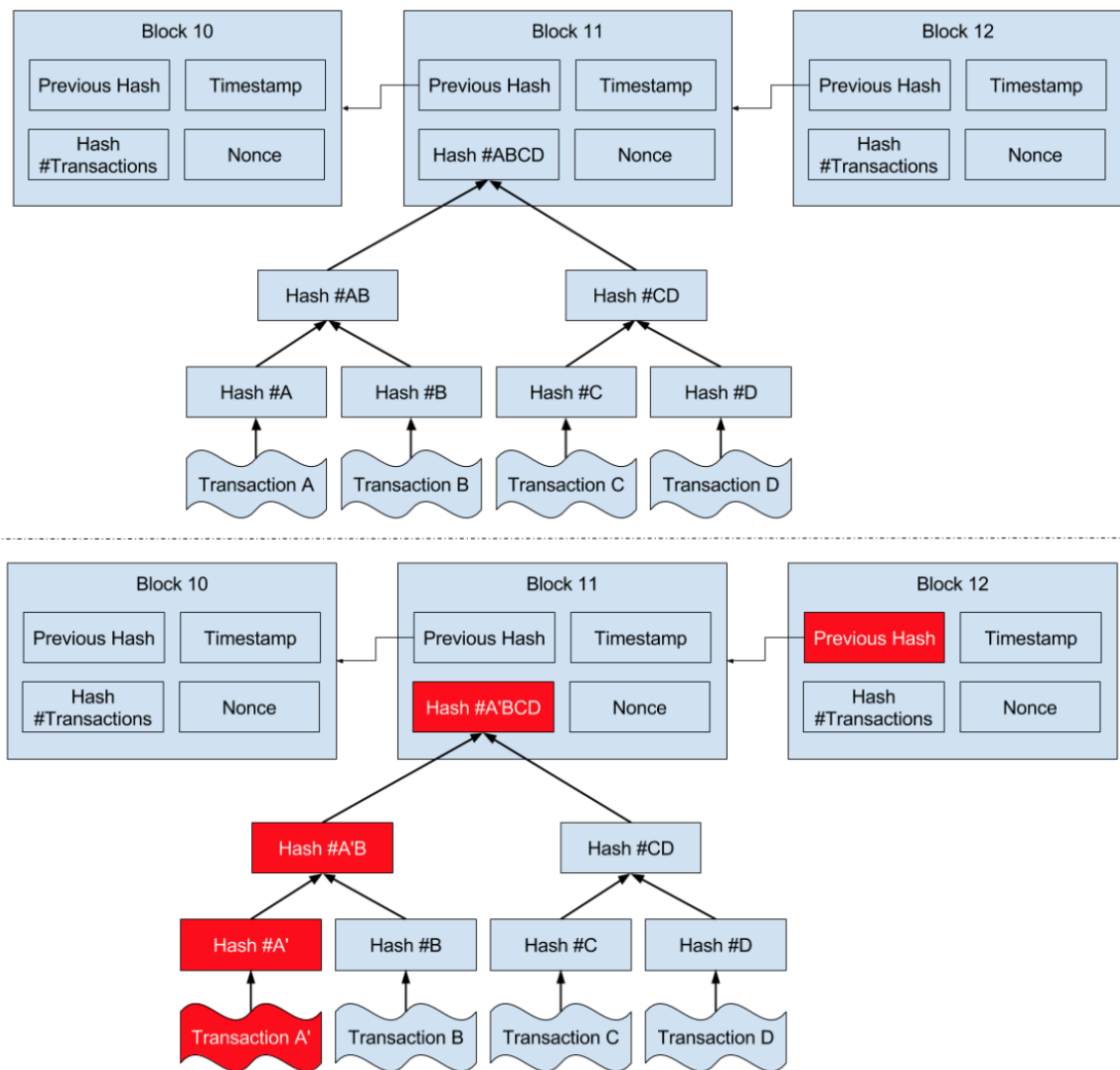


Figure 1.3 – Blockchain immutability

1.2.4 Hash functions. Hashing is the process of converting an array of input data of arbitrary length into an (output) bit string of fixed length. For example, a hash function can take a string with any number of characters (one letter or a whole literary work), and get a string with a strictly defined number of characters (digest) at the output. Hash functions are available in almost any programming language. A separate category of hash functions is cryptographic hash functions. They are subject to much more stringent requirements than to the functions commonly used in hash tables. Therefore, they are used in more "serious" cases, for example, for storing passwords. Cryptographic hash functions are developed and thoroughly checked by researchers around the world [9].

To continue to function, the block must create new blocks. Since blockhouses are decentralized systems, new blocks should be created not by the only authenticating entity, but by the network as a whole. To decide what the new block will be, the network must reach a consensus. To reach a consensus, the miners offer certain blocks, the blocks are verified, and finally the network selects the only block that will be the next part of the registry. However, many miners offer identical blocks that pass verification.

Hash functions in lockers guarantee the "irreversibility" of the entire transaction chain. The matter is that each new block of transactions refers to the hash of the previous block in the registry. The hash of the block itself depends on all the transactions in the block, but instead of sequentially passing the hash function transactions, they are collected in one hash value using a binary tree with hashes (the Merkle tree). Thus, hashes are used as a replacement for pointers in conventional data structures: linked lists and binary trees.

The Merkle tree, likewise referred to as a binary hash tree, is actually an information system which is actually utilized to hold hashes of the person information in big datasets in a method to come up with the verification of the dataset effective. It's an anti tamper mechanism to make sure that the larger dataset hasn't been modified. The word 'tree' is actually utilized to relate to a branching information system in computer science, as observed in the picture below [10].

Hash tasks are available for sale in nearly every programming language. For instance, they're employed to execute hash tables as well as sets (HashMap / HashSet in Java, set and dict in Python, Map, Objects and Set in JavaScript, and so on). A specific category of hash tasks is cryptographic hash features. They're subject to far more stringent requirements than to the features widely used in hash tables. Thus, they're used in more "serious" cases, for instance, for saving passwords. Cryptographic hash tasks are created and completely examined by researchers around the globe..

Because of the usage of hashes, the normal state of the blocking system - all of the transactions which have been carried out and the sequence of theirs - will be conveyed by an one-time number: the hash of the most recent obstruct. Thus, the invariance property of a hash of 1 obstruct promises the invariability of the whole block.

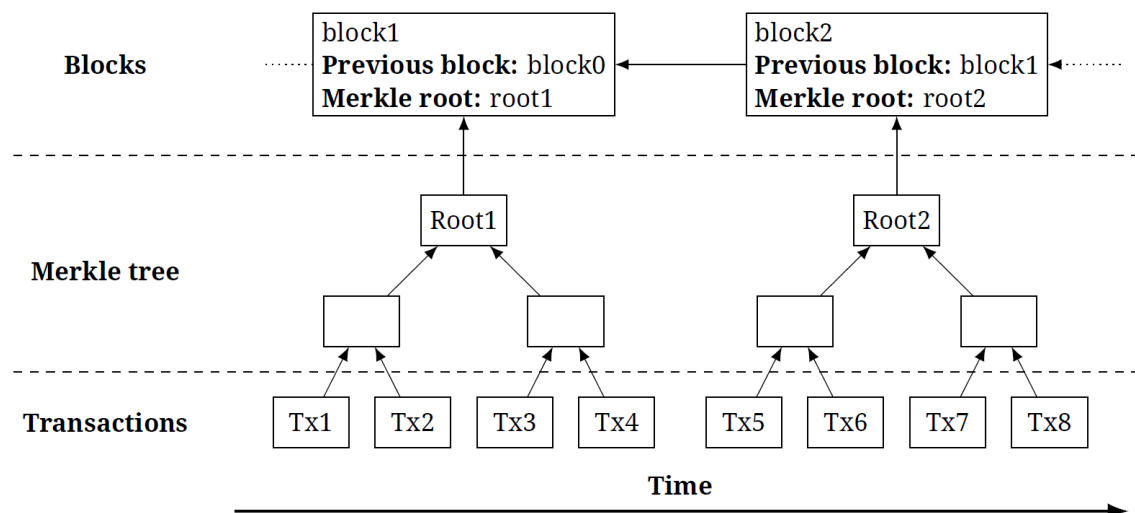


Figure 1.4 - Block Hashing

Timestamping is another key feature of blockchain technology. Each block is timestamped, with each new block referring to the previous block. Combined

with cryptographic hashes, this timestamped chain of blocks provides an immutable record of all transactions in the network, from the very first (or genesis) block.

1.2.5 Digital Signatures. Digital signatures in block systems are based on public key cryptography. It uses two keys. The first - a private key - is needed to generate digital signatures and is kept secret [11]. The second - the public key - is used to verify the electronic signature. The public key can really be calculated on the basis of a private key, but the reverse transformation requires an impossible amount of computation in practice comparable to brute force.

There are many different public-key cryptography schemes. The two most popular of these are the schemes based on factoring (RSA) and schemes based on elliptical curves. The latter are more popular in blockhouses due to the smaller size of the keys and signatures. For example, in bitcoin, the ECDSA elliptical cryptography standard is used, together with the elliptic curve secp256k1. In it, the private key is 32 bytes long, open - 33 bytes, and the signature - about 70 bytes.

1.2.6 Private and public keys. A private key is generated by the user himself, used to sign transactions. It is kept secret, the one who owns the private key has access to the lockbox cell, which can be represented by a purse, a container with any data (for example, personal correspondence, important documents, etc.).

An open (public) key must be generated on the basis of a private key, that is, there is a mathematical connection between them (the public key is not invented from the head). It can be published, moreover, it is used in the detachment as a block address, and also as a verification of the authenticity of the signature of information in other blocks, by third-party network members. Knowledge of the public key does not make it possible to determine the private key [14].

To create a signature, you will need:

- asymmetric encryption algorithm (for example, RSA);
- hash function (for example, SHA512);
- information that we are going to sign.

Since asymmetric algorithms are quite slow compared to symmetric algorithms, the amount of data to be signed plays a large role and if it is large, then they usually take a hash from the data being signed, not the data itself. The hash is obtained using hash functions, for example, SHA512, which accepts some information on the input and returns a hash of a certain length. Hash function as a meat grinder, you can scroll the meat and get minced, but back from the stuffing already meat will not get. Thus, the EDS is placed not on the document itself, but on its hash. Hash functions are not part of the EDS algorithm, so any reliable hash function can be used in the scheme [15].

Stages:

- 1) using RSA, generate a pair of public and private keys;
- 2) the signed data is substituted into function SHA512 and we get a hash;
- 3) put the received hash and private key into the function of asymmetric RSA encryption, that is, RSA Encode (hash of information, private key), we get the line - EDS on the output.

1.2.7 Consensus Algorithms. Consensus in the networking refers to the procedure of obtaining agreement with the networking participants as to the appropriate status of details on the product. Consensus results to other nodes sharing the very same details. A opinion algorithm, hence, does 2 things: it guarantees that the information on the ledger is actually the exact same for all of the nodes within the network, as well as, in turn, stops malicious actors from adjusting the information. The opinion algorithm varies with various blockchain implementations.

Even though the Bitcoin blockchain makes use of Proof of Work because the opinion algorithm, additional blockchains and distributed ledgers are actually deploying a bunch of consensus algorithms, similar to the Proof of Stake, Proof of Burn, Proof of Capacity, Proof of Elapsed Time, as well as a lot others, based on the unique needs of theirs.

1.2.8 Proof of Work (PoW). The Proof of Work opinion algorithm consists of solving a computational demanding puzzle to be able to produce brand new blocks within the Bitcoin blockchain. Colloquially, the task is actually recognized as ' mining', as well as the nodes in the network which take part in mining are actually widely known as ' miners'. The motivator for mining transactions lies to come down with economic payoffs, exactly where competing miners are actually compensated with 12.5 bitcoins and a tiny transaction fee[12].

Several criticisms can be found for the PoW opinion algorithm. PoW calls for a large quantity of electricity to be expended, provided the computationally major algorithm. Additionally, PoW has an excessive latency of transaction validation, as well as the focus of mining power is actually put in lands where electricity is actually inexpensive. As for the network safety measures, PoW is actually vulnerable to the ' fifty one % attack', that typically refers to an assault on a blockchain by a team of miners controlling greater than fifty % of this network's computing energy.

1.2.9 Proof of Stake (PoS). The Proof of Stake algorithm is actually a generalization on the Proof of Work algorithm. Inside PoS, the nodes are actually referred to as the ' validators' and even, instead of mining the blockchain, they validate the transactions to generate a transaction fee. There's no mining to be completed, as almost all coins are available from day one. To put it simply, nodes are arbitrarily selected to verify blocks, as well as the likelihood of this arbitrary choice is dependent on the quantity of stake held. Thus, in case node X owns two coins as well as node Y owns one coin, node X is two times as apt to be called upon to verify an obstruct of transactions. The particular implementation of PoS is able to vary, based on the usage case, or perhaps as a situation of software design. Instances include Proof of Proof as well as Deposit of Burn. The PoS algorithm saves costly computational energy which are invested in mining within a PoW opinion regime .

1.2.10 Proof of Elapsed Time (PoET). Created by Intel, the Proof of Elapsed Time opinion algorithm emulates the Bitcoin style Proof of Work. Hyperledger's Sawtooth implementation is actually a good example of PoET at the office. Rather than competing to resolve the cryptographic headache and then mine the following obstruct, as inside the Bitcoin blockchain, the PoET opinion algorithm is actually a cross types of an arbitrary lottery as well as first-come-first-serve foundation. Inside PoET, each validator is actually provided an arbitrary wait time [16].

1.3 Smart Contracts

Smart Contracts are simply computer programs that execute predefined actions when certain conditions within the system are met. Smart contracts provide the language of transactions that allow the ledger state to be modified. They can facilitate the exchange and transfer of anything of value (e.g. shares, money, content, property).

To be able to conclude a typical offer, you have to visit a lawyer or perhaps a notary, pay and hang on for the files to be given. Sensible contracts job like vending machines: you just throw bitcoin into the machine (which is actually, in the registry), as well as the agreement maintained by a third get-together, driver's license or maybe some additional service you purchased falls into the bank account of yours.

Additionally, not like conventional agreements, sensible contracts not just include info regarding the parties' penalties as well as responsibilities for the violation of theirs, but additionally instantly guarantee the fulfillment of all of the terms of the agreement [13].

Eventually, this particular system confirms the fulfillment of the terms of the agreement and instantly determines if the specified advantage ought to go to one of the participants in the transaction or perhaps quickly go back to another participant (and maybe the circumstances are somewhat far more complicated). All of this time the document is actually saved as well as duplicated in the decentralized registry, which guarantees the reliability of its and doesn't let any of the parties to alter the terms of the agreement.

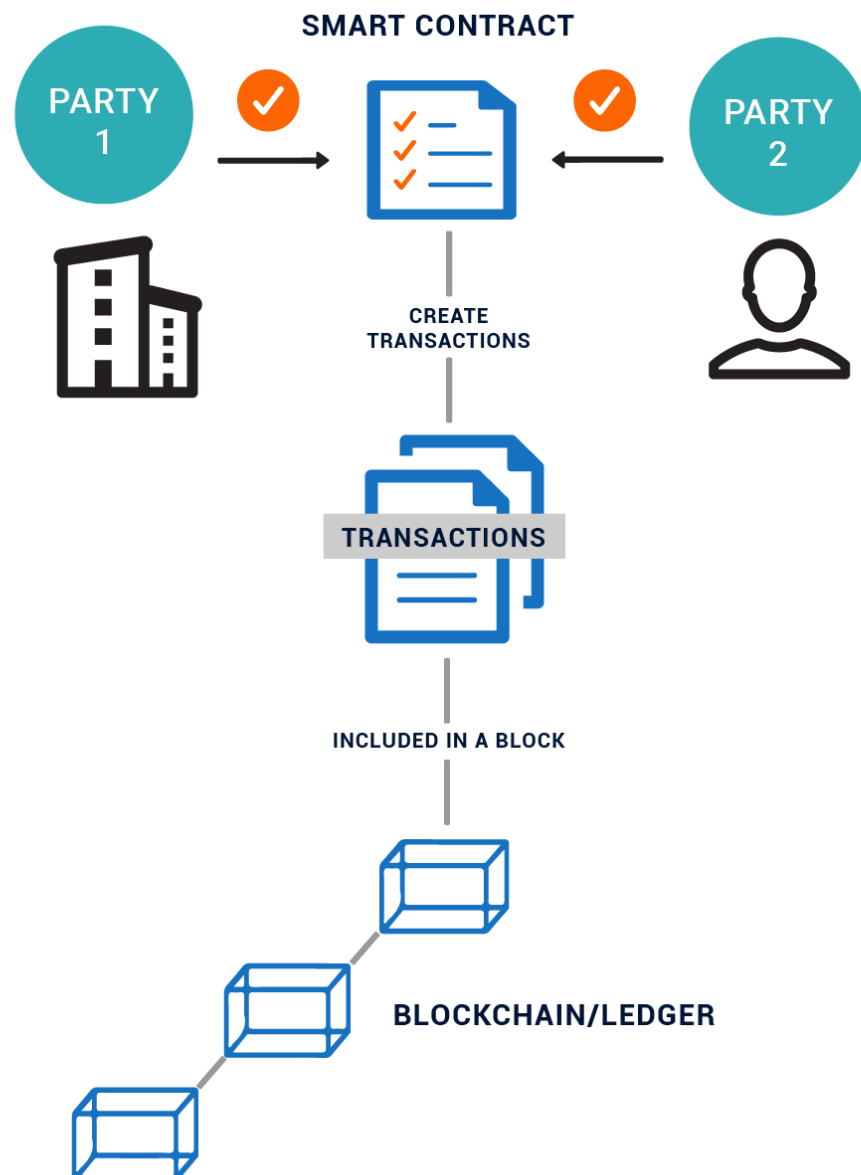


Figure 1.5 - Structure of smart contracts

When we talk about distributed computing, the load distribution is usually understood. Let's say we have a task to calculate the occurrence of a specific word in a huge text file. We cut the file into several parts, we distribute these parts to different nodes (for example, using Hadoop), each performs a count and produces a response. We summarize the answers and get the result. Thus, we significantly speed up the task[17].

If we talk about smart contracts, then this is a completely different situation. Here we do not cut the file into separate parts, we give each node a whole file, and each node gives us the same result (ideally). Let's return to our question: does such an action fall under the definition of distributed computing? Well, in general, why not? They are distributed and they are the same calculation? Only in this case we are not talking about the distribution of the load, but about the distribution of trust.

And yes, it should be noted that such a concept is difficult to scale, because it does not initially contain the necessary mechanisms. Why such a construction? The most obvious example is the creation of a social contract (contract) between two or more parties. Hence the name of the concept - "smart contract". The parties fix the arrangements and conditions for the fulfillment of a scenario for the development of relations among themselves, using a programming language, in such a way that the occurrence of certain events will automatically cause the execution of a predefined code.

A smart contract is a condition written in computer language, in which the parties signing a smart contract exchange any assets: currency, real estate, shares, etc. For example, the buyer's currency is transferred to the program and frozen there until , until the seller fulfills its part of the contract. If the condition is disrupted, the amount is returned to the client's account, and the smart contract is canceled. If all conditions are fulfilled, then an asset exchange occurs [18]. This exchange is fixed in a smart contract and is written to the block account, after which it can not be canceled or replaced or destroyed. By tracking the fulfilled conditions the program is engaged in an automatic mode, control or participation of people is not needed. In other words, smart contracts work directly between stakeholders, excluding intermediaries.

Smart contracts provide an opportunity to safely exchange money, shares, property and other assets directly, without intermediaries.

In order to conclude any transaction, you need to contact a notary or lawyer, pay for the documents and wait for their registration. Often, many clauses of these documents contain references to legislative articles that can be interpreted by themselves, bypassed. In the event of non-fulfillment of the terms of the transaction, in real life people have to go to court, again spend money on the process and prove their case. At the conclusion of such transactions in general there can be no talk about the trust of the parties to the treaty.

Smart contracts can regulate a variety of financial (and not only) relationships between people. The most obvious option is trading on the Internet. E-commerce covers today almost all types of goods. We order not only machinery, but also ready meals, products.

Above we have already given an example with the purchase of real estate. Let's figure out how you can implement the option with its lease. We must deposit money for the first month of rent and a deposit. The amount is fixed in the blockhouse, after which the landlord hands over the keys.

To fully automate the smart contract, you need to add a little "Internet of things": it is advisable to install a heaped lock in a removable dwelling, which will be automatically blocked if the payment is late or at the end of the agreed period. When the lease comes to an end, the doors will be blocked and the tenant frozen automatically returns to the tenant.

In addition, smart contracts can be used in the distribution of inheritance. An elderly billionaire who does not trust executors (the human factor, billions of inheritance - you understand), prescribes in the smart contract the accounts of the beneficiaries of the inheritance in case of his death. The system periodically tracks

information from the state registry of the deceased. As soon as there appears a record about the wanted billionaire, the money is automatically sent to his happy heirs.

1.4 IOTA

For the first time, the crypto-currency community learned about this currency in 2015, and all this time the developers were engaged in bringing their ideas to mind, and they look like this. Crypto currency should not be controlled by anyone. Users of cryptolayout should be able to conduct micropayments. Absence of commission for conducted transactions, and speed and conduct - this is one of the most important criteria. All these ideas are implemented in the blockchain iota. This crypto currency has another very interesting function, in order for you to conduct a transaction, it must be confirmed by other network members (3 people), and the sooner you receive this confirmation, the faster you will make a transaction. People who confirm your transaction are waiting for their confirmation, which allows you to make quick transactions. Such a system can be used not only in the sale of things on the Internet. If you do not know, then IOTA was originally conceived as a crypto currency for the Internet of things. This principle can be used in voting and other spheres [19].

Basically, the platform involves a generalization of this blockchain protocol (the engineering known as Tangle) which sits at the backend on the IOTA platform.

Rather than paying miners to verify the transactions, the structure of the networking entails peer based validation. We are able to think of an easy analogy, that of a mentor grading students' homework: the pupils are actually the clients/users within the Bitcoin process, as well as the teacher is actually the miner/validator. Tangle engineering asks pupils (users) to quality every other's research, generating the demand for an instructor (external validator) unwanted, along with staying away from expenditures connected to the teacher's/validator's deliver the results. This enables the platform to become totally free of price, without dealing with the scaling problems that are actually natural in the very first generation of blockchains.

Additionally, the use of the platform with connected devices or the Internet of Things.

Many people believe that the principle of IOTA's work has solved all the problems of bitcoin and this system is even more promising. We will not be so categorical, but we will list the main advantages of IOTA before bitcoin. There is no division and privileges in the network - there are no ordinary participants and those who confirm transactions. All users are absolutely equal and nobody depends on anyone. IOTA is an absolutely decentralized system. In the same bitcoin, the miners can assemble into pools (which happens) and affect the network (which also happens). In the IOTA, this is simply impossible.

There are no commissions in the IOTA network at all. Again, because there are no miners. Therefore, the system is ideal for micropayments and has no

restrictions on their volume and amount. In fact, it is even better promoted by Ripple or Stellar, which position themselves as payment solutions.

All networks such as bitcoin have problems with scalability, which we already observe. The unique algorithm used by IOTA solves this problem almost forever. Here, on the contrary - the more participants, the better the system works.

Using the different operating principles of the Proof-of-Work and Proof-of-Stake systems reduces the hardware requirements. For IOTA enough conventional devices that people use, and which most. In technical terms, a fairly inexpensive microcontroller with 16 kb of RAM. IOTA uses a different type of encryption. Without going into details, this is the principle of quantum proof, which is designed and tuned to quantum computers. And although the latter have not yet appeared, but this is the prospect of the near future.

1.5 Ethereum Smart Contracts

A hypothetical example of an Ethereum based wise contract might entail all of the following transaction: within an equity raise, transfer quantity X in the investor to the organization upon getting the given shares coming from the business. The monetary quantity X, that had been pre validated by way of the organization with the transaction (much love in a charge card purchase), is actually kept in escrow by the intelligent agreement, until the shares are obtained by the investor. Any sort of arbitrary advanced business logic could be dedicated to the blockchain. The Ethereum blockchain just encodes these rules belonging to the games'. The particular payoffs happen by mingling with the blockchain [21].

The illustration beneath describes this procedure. The sensible agreement rests on the Ethereum public blockchain, and it is operated on the Ethereum Virtual Machine (EVM). Once punching in a triggering occasion, as an expiration date or maybe a hit price which has been pre coded, the smart agreement instantly executes as per the online business logic As an additional benefit, regulators can study the market activity on a continuing foundation, without the need of compromising the identity of certain players inside a permissionless public blockchain, as Ethereum.

Sensible contracts are the key component of Ethereum. From them any algorithm may be encoded. Sensible contracts are able to carry arbitrary condition and will conduct almost any arbitrary computations. They're actually in the position to call different wise contracts. This provides the scripting facilities of Ethereum huge flexibility.

Sensible contracts are actually run by each node together with the block development system. Much love Bitcoin, block development is actually the moment where transactions really take place, inside the sense that the moment a transaction takes place within a block, worldwide blockchain express is actually modified. Buying impacts state changes, and the same as in Bitcoin, every node is actually no cost to select the order of transactions within a block. Right after doing very (and performing the transactions), a specific amount of work should be performed to produce an appropriate block. In comparison to Bitcoin, Ethereum uses an alternative pattern for choosing which blocks get extra to the appropriate

blockchain. While in Bitcoin probably the longest chain of legitimate blocks is definitely the rightful blockchain, Ethereum uses a process known as GHOST (in reality a deviation thereof) [20]. The GHOST protocol enables for stagnant blocks, blocks which were computed by some other nodes but that could usually be thrown away because others have computed more recent blocks, to be incorporated into the blockchain, decreasing squandered raising rewards and computing power for slower nodes. Additionally, it allows for quicker confirmation of transactions: while in Bitcoin blocks usually are produced every ten minutes, within Ethereum blocks are made in just seconds. Much conversation has gone into if this particular process is an enhancement with the easier "fastest lengthiest chain" process for bitcoin, however this particular dialogue is out of range for this document. For today this process seems to work with results in Ethereum.

A crucial facet of just how smart contracts job of Ethereum is they've the own address of theirs of the blockchain. Put simply, contract code isn't carried within each transaction which uses it. This will rapidly be unwieldy. Rather, a node is able to make an unique transaction which assigns addresses to a contract. This particular transaction also can run code at the second of creation. Once this first transaction, the agreement becomes permanently a component of the blockchain as well as the address of its never changes. Anytime a node wishes to call up any of the techniques identified by the agreement, it is able to send an email to the address with the contract, specifying information as input and also the technique which should be called. The contract is going to run as part of the development of more recent blocks up to the gasoline restrict or perhaps conclusion [24]. Contract techniques are able to return a value or even store information. This particular information is an element of the state of this blockchain.

1.5 Hyperledger

Hyperledger is actually an open source work intended to advance cross industry blockchain technologies. Hosted by The Linux Foundation, it's a worldwide effort of members from different industries & organizations. Hyperledger boasts a multitude of enterprise ready remedies. Hyperledger is all about towns of software designers building blockchain frameworks as well as platforms [22].

Hyperledger has a substitute to the cryptocurrency based blockchain version, as well as concentrates on developing blockchain frameworks as well as modules to allow for worldwide enterprise solutions. The emphasis of Hyperledger is actually providing a collaborative and transparent strategy to blockchain development.

Hyperledger business blockchain frameworks are utilized to develop business blockchains for a consortium of businesses. They're completely different compared to public ledgers including the Bitcoin blockchain as well as Ethereum. The Hyperledger frameworks include:

- an append only distributed ledger;
- an opinion algorithm for agreeing to switches of the ledger;

- privacy of transactions by way of permissioned access; - shrewd contracts to thing to do transaction requests.

1.5.1 Hyperledger Iroha. Hyperledger Iroha is actually a blockchain framework contributed by Colu, Hitachi, and Soramitsu. Hyperledger Iroha is actually created to be easy and simple to integrate into infrastructure projects needing sent out ledger technologies. Hyperledger Iroha focuses on mobile program development with customer libraries for Ios as well as Android, making it unique from some other Hyperledger frameworks. Inspired by Hyperledger Fabric, Hyperledger Iroha seeks to enhance Hyperledger Fabric as well as Hyperledger Sawtooth, while offering a development setting for C++ designers to add to Hyperledger [22].

To conclude, Hyperledger Iroha includes a basic construction, modern, domain driven C layout, together with the consensus algorithm YAC.

1.5.2 Hyperledger Sawtooth. Hyperledger Sawtooth, contributed by Intel, is actually a blockchain framework that employs a modular platform for constructing, deploying, and working distributed ledgers. Distributed ledger treatments designed with Hyperledger Sawtooth is able to make use of many consensus algorithms based on the dimensions of the network. By default, it utilizes the Proof of Elapsed Time (PoET) consensus algorithm, which supplies the scalability on 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 versatility, with assistance for each permissioned as well as permissionless deployments.

Sawtooth is actually a modular platform provided by Intel found April 2016, with a few main innovative developments. The concentration is actually on adaptable use in a variety of business places, with the launch of other consensus and transaction families. Transactions are actually separated from the consensus degree, making use of 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 [23]. 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.

1.5.3 Hyperledger Fabric. Hyperledger Fabric was the first proposition for a codebase, merging earlier work carried out by Digital Asset Holdings, Blockstream's libconsensus, as well as IBM's OpenBlockchain. Hyperledger Fabric has a modular architecture, that enables elements including consensus as well as membership solutions to be plug-and-play. Hyperledger Fabric is groundbreaking

in allowing entities to carry out confidential transactions without passing info by way of a main authority. This's achieved through diverse stations which operate to the network, in addition to the division of labor which characterizes the various nodes to the network. Finally, it's essential to recall that here, unlike Bitcoin, that is a public chain, Hyperledger Fabric supports permissioned deployments.

1.5.4 Hyperledger Burrow. Formally recognized as eris-db, Hyperledger Burrow was launched in December 2014. Presently below incubation, Hyperledger Burrow is actually a permissionable smart contract machine which offers a modular blockchain prospect with a permissioned wise contract interpreter built in deep portion to the specification on the Ethereum Virtual Machine (EVM). It's the one available Apache licensed EVM implementation [22].

Below are the main ingredients of Burrow:

- the gateway supplies interfaces for methods integration as well as pc user interfaces;
- the smart contract the application motor facilitates integration of complicated internet business logic;
- the consensus engine;
- Application Blockchain Interface (ABCI) offers with the assistance of Burrow user interface specification for the consensus engine as well as sensible contract program engine to connect [25].

2 Practical realization of the project

2.1 Formulation of the problem

One of the main reasons why a blockchain is so attractive is that the chains almost always have open source code. This means that other users or developers have the ability to change it at their discretion, but at the same time it makes it incredibly difficult to imperceptibly change previously registered data. The latter circumstance makes the blockade a particularly reliable technology.

Advantages of blockchain technology:

- the block database is safe, unlike the classical database where information is stored on the server, and all people with access have the ability to destroy, steal or use data.
- another feature of the blockchain is openness. In technology, each node independently verifies information and has the ability to perform a transaction. The block chain is characterized by the transparency of the transmitted data.

Another serious reason for the attractiveness of the block is that the technology does not involve a central point of data collection. Instead of running a large data center and conducting all transactions through it, the blocking system actually allows individual transactions to have their own authentication and authorization to ensure their communication with each other. Information about specific blocks of the chain is scattered across different servers around the world, and this ensures that even if this information gets to outsiders, for example,

hackers, only a small amount of data will be compromised, not the whole network [24].

Total as a platform for the project was chosen Hyperledger Fabric. The advantages of this platform will be disclosed below. The formulation of the tasks is as follows:

- studying the architecture of the selected platform;
- study of the process of accepting and processing transactions;
- making changes to the program code for the implementation of the student's progress record application;
- making changes in the the client side of the application
- calculation of block creation time
- calculation of the number of transactions within the block;
- fault tolerance calculation;
- planning for further development of the application.

2.2 Justification of the choice of Hyperledger Fabric

Hyperledger is a Linux foundation product that was created in December 2015 to promote the technology of the blockchain. The project is a joint effort to create an open, distributed accounting system (ledger), which can be used to openly develop and implement blockchain applications and systems. The main emphasis is on creating and launching platforms that support global business transactions. The project also focuses on improving the reliability and productivity of the detachment.

Projects in Hyperledger go through various stages of development: from the idea and its development to active work and even stagnation. In order for a project to move from an idea to an incubation stage, it must have a fully functioning code base and an active development community.

Fabric is a blockchain framework that was originally proposed by IBM and DASH (Digital Asset Holdings). It is designed to create the basis for developing solutions for the blockchain and is based on a modular architecture, where, if necessary, various components can be connected, for example, a consensus algorithm. Smart contracts in Fabric are called chaincode. The Hyperledger Fabric network includes "peer-to-peer nodes" that perform block code, access register data, support transactions, and an application interface. Fabric is designed for projects that require distributed register technology (DLT). Supports chaincode in Go (Golang), Java and JavaScript (via Hyperledger Composer, or from version 1.1) and therefore potentially more flexible than the language of smart contracts. Hyperledger Fabric is a platform for distributed ledger solutions, underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility and scalability. It is designed to support pluggable implementations of different components, and accommodate the complexity and intricacies that exist across the economic ecosystem.

Hyperledger Fabric delivers a uniquely elastic and extensible architecture, distinguishing it from alternative blockchain solutions. Planning for the future of

enterprise blockchain requires building on top of a fully-vetted, open source architecture [25].

2.2.1 Hyperledger Fabric Functionalities. Hyperledger Fabric is actually an implementation of distributed ledger technology (DLT) which provides enterprise ready network security, scalability, performance & confidentiality, in a modular blockchain architecture. Hyperledger Fabric provides the following blockchain networking functionalities: identity management, confidentiality and privacy, effective processing, chaincode performance, modular style.

In order to allow permissioned networks, Hyperledger Fabric has a membership identity service which manages user IDs as well as authenticates all participants on the network. Access management lists can be utilized to supply extra levels of permission via authorization of certain community operations. For instance, a certain user ID may be allowed to invoke a chaincode program, but blocked from deploying fresh chaincode.

Hyperledger Fabric allows competing company interests, and any groups which need private, confidential transactions, to coexist on the same permissioned networking. Private channels are actually restricted messaging paths which may be utilized to give confidentiality as well as transaction privacy for certain subsets of network users. All information, which includes channel information, member, and transaction, on a channel are actually inaccessible and invisible to any kind of network participants not explicitly given access to that channel.

Hyperledger Fabric assigns networking roles by node sort. In order to give parallelism and concurrency to the network, transaction delivery is actually separated from transaction ordering and commitment. Executing transactions just before purchasing them allows each peer node to process several transactions concurrently. This concurrent execution increases processing effectiveness on each peer along with accelerates delivery of transactions to the buying service. Along with enabling parallel processing, the division of labor unburdens purchasing nodes from the challenges of transaction delivery as well as ledger maintenance, while peer nodes are actually freed from ordering (consensus) workloads. This particular bifurcation of roles likewise limits the processing necessary for authentication and authorization ; all peer nodes don't need to believe in everything ordering nodes, and vice versa, therefore tasks on one may operate independently of verification by the other hand [26].

Chaincode programs encode logic which is actually invoked by certain kinds of transactions on the channel. Chaincode which defines parameters for a difference of asset ownership, for instance, guarantees that all transactions that transfer ownership are actually subject to exactly the same rules & requirements. Method chaincode is actually distinguished as chaincode which defines operating parameters for the whole channel. Lifecycle as well as configuration method chaincode describes the rules for the channel; endorsement as well as validation method chaincode defines the demands for endorsing as well as validating transactions.

Hyperledger Fabric tools a modular architecture to offer purposeful decision to community designers. Certain algorithms for identity, ordering (consensus) as

well as encryption, for instance, may be plugged in to any Hyperledger Fabric network. The effect is a common blockchain architecture that public domain or just about any market is able to adopt, with the guarantee that the networks of its will likely be interoperable across marketplace, geographic and regulatory boundaries.

You will find 3 distinct kinds of roles inside a Hyperledger Fabric network:

- consumers are actually uses which act on behalf of an individual to propose transactions on the network;

- peers maintain the state of a copy along with the network of the ledger.

You will find 2 various kinds of peers: committing as well as promoting peers. Nevertheless, there's an overlap between endorsing as well as committing peers, in that endorsing peers are actually a specific type of committing peers. All peers commit blocks to the distributed ledger: endorsers simulate as well as endorse transactions, Committers verify recommendations and validate transaction benefits, just before committing transactions to the blockchain;

- the ordering service accepts endorsed transactions, orders them right into a block, as well as provides the blocks to the committing peers.

A channel is a private subnet between two or more network members for conducting confidential transactions. It performs the following functions:

- registries exist within a channel;
- registries can be distributed among all or specific network nodes;
- peer can be connected to multiple channels;
- chaincode is installed on peers that need access to data in the registry;
- chaincode instance is created within a specific channel with an endorsement policy;
- the state created by chaincode is available only to him.

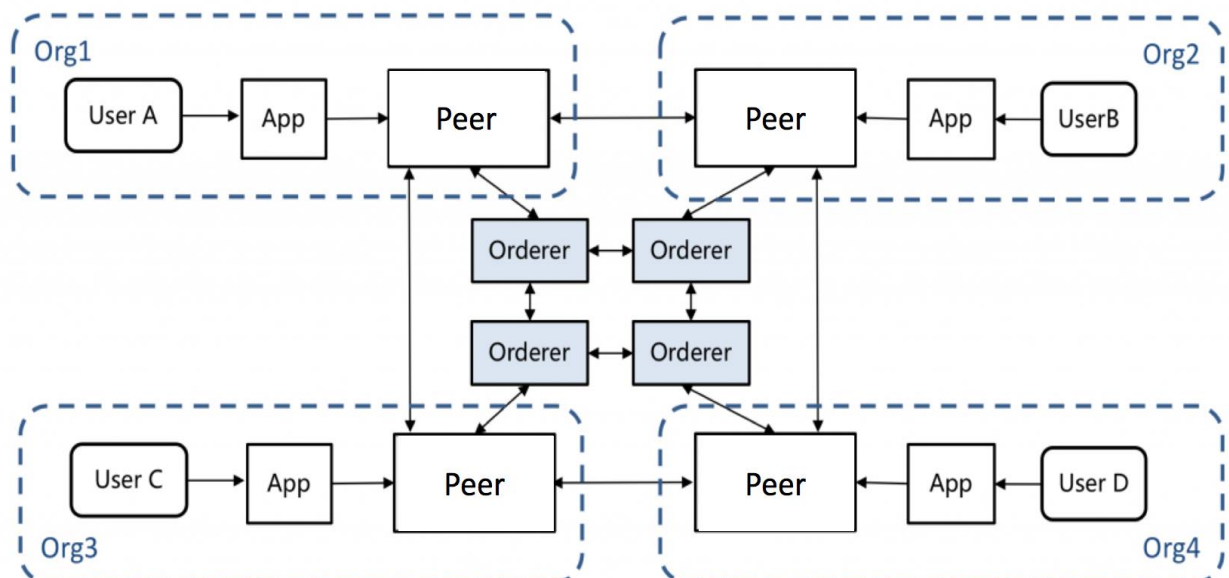


Figure 2.1 – The role of participants in the Hyperledger Fabric network

As can be seen from the diagram - each participant has:

- its own node (peer), which ensures the health of the block network and is part of the HLF;

- an application that usually consists of at least a web server and a user interface;
- an ordering service, which provides interaction between nodes [23].

In the application, users are authorized with their own keys or a binder login and password and have access to a smart contract.

A smart contract is a programmed business logic and rules for interacting with data that are placed in a distributed ledger. The registry contains a chain of transaction blocks that members of the network have committed. Each subsequent block stores the hash of the previous block, which ensures that some of the information can not be changed. The process of agreeing the correctness of a block and adding it to a distributed registry is called consensus. Different block platforms have different consensus mechanisms: proof of work (PoW), proof of stake (PoS), byzantine fault tolerance (BFT), and others [23].

Beginning with version 1.0, HLF is specific for the presence of the Ordering Service, which is distributed among the participants in the cluster and is responsible for the order of transactions in the forming block. According to the configured parameters, it collects transactions on the network and forms a block. In the event of the failure of the Ordering Service, transactions on the network will cease to be registered, but the data itself will remain unchanged.

2.3 Transaction Flow

Within a Hyperledger Fabric network, transactions begin with client apps driving transaction proposals, or even, in other words, proposing a transaction to endorsing peers.

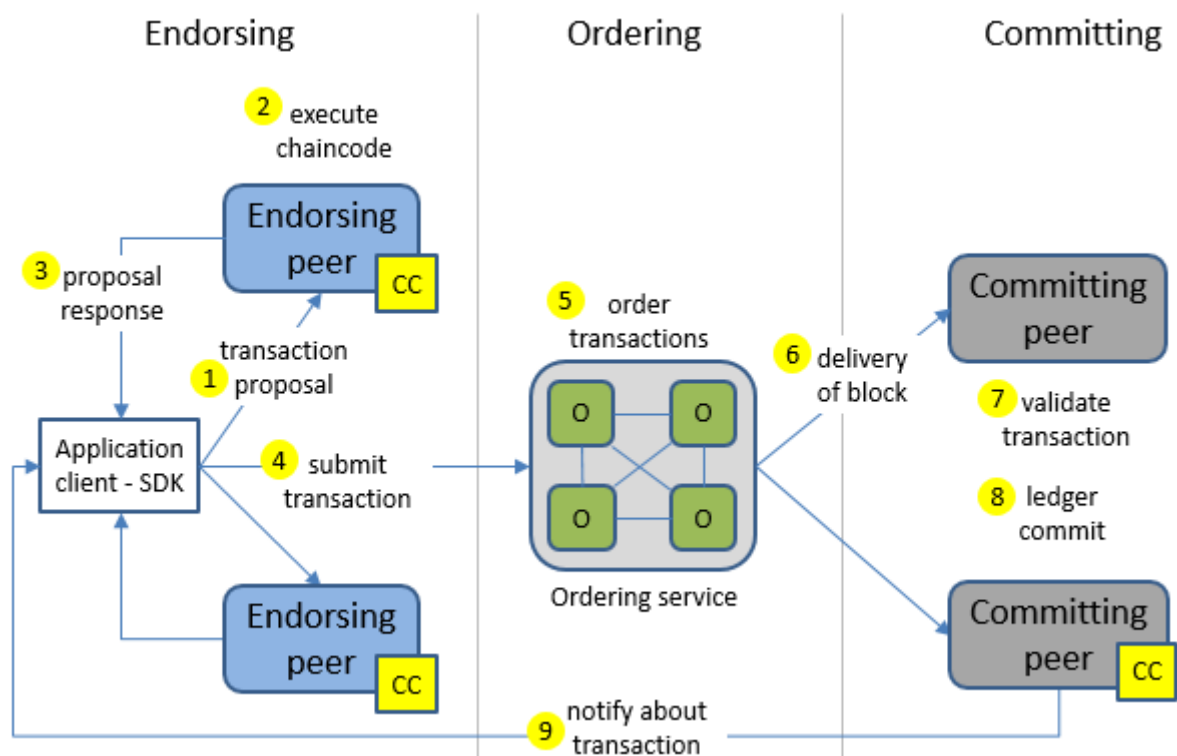


Figure 2.2 – The role of participants in the Hyperledger Fabric network

Client programs are known as customers or applications, as well as enable individuals to speak with the blockchain network. Program developers could use the Hyperledger Fabric networking with the application SDK [24].

Each promoting peer simulates the suggested transaction, without the need of updating the ledger. The endorsing peers are going to capture the set of Read and also Written information, known as RW Sets. These RW sets capture that which was read out of the present world express while simulating the transaction, along with what would've been created to the planet declare had the transaction been performed. These RW sets are then signed through the promoting peer, along with refunded to the prospect program to be utilized in future measures of the transaction flow.

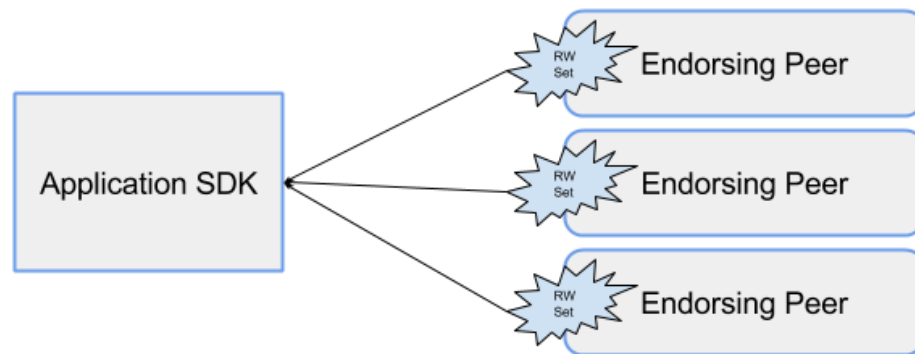


Figure 2.3 – Endorsers simulate transactions

Endorsing peers should hold smart contracts to imitate the transaction proposals.

A transaction endorsement is actually a signed reaction to the outcomes of the simulated transaction. The strategy of transaction endorsements is dependent on the endorsement policy that is actually specified once the chaincode is actually deployed. A good example of an endorsement policy will be "the bulk of the endorsing peers should endorse the transaction". Because an endorsement policy is actually specified for a certain chaincode, various stations are able to have various endorsement policies [17].

The software then submits the backed transaction and also the RW sets to the buying service. Buying takes place throughout the networking, in parallel with backed transactions as well as RW sets submitted by various other applications.

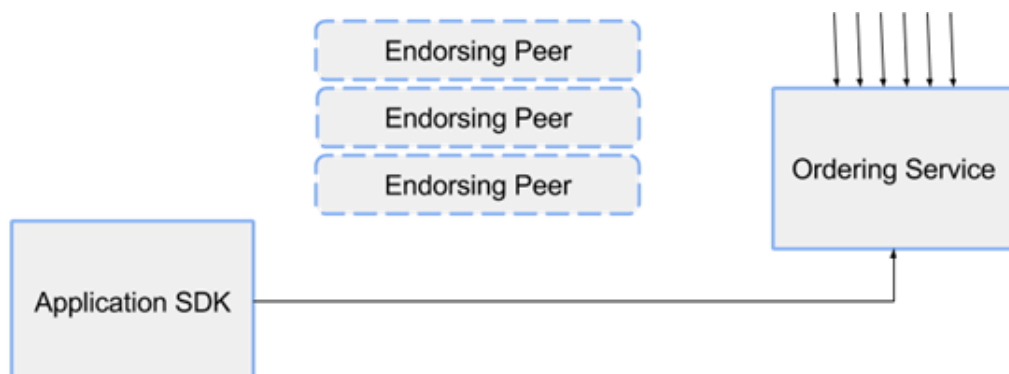


Figure 2.4 – Client application submits to ordering service

The ordering service takes the endorsed transactions and RW sets, orders this information into a block, and delivers the block to all committing peers.

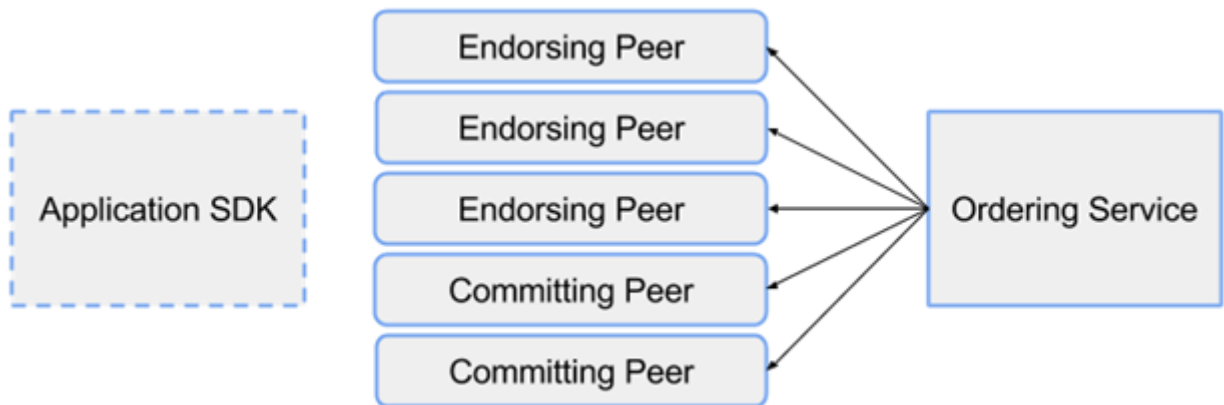


Figure 2.5 – Orderer sends ordered transactions in a block to all peers

The ordering service, which is comprised of a bunch of orderers, doesn't process transactions, sensible contracts, or perhaps keeps the shared ledger. The buying system accepts the backed transactions and says the order in which people transactions will be dedicated to the ledger. The Fabric v1.0 structure continues to be created so that the particular setup of 'ordering' (Solo, Kafka, BFT) gets a pluggable component. The default buying service for Hyperledger Fabric is actually Kafka. Thus, the buying service is a modular part of Hyperledger Fabric [28].

It's crucial that you be aware that the state of the networking is actually maintained by peers, moreover not by the buying service or maybe the client. Usually, you are going to design the program of yours so that various devices in the network play various roles. That's, devices which are a part of the buying service shouldn't be set up to additionally endorse or even commit transactions, and the other way round. Nevertheless, there's an overlap between endorsing as well as committing peers on the product. Endorsing peers should get access to and hold sensible contracts, additionally to fulfilling the job associated with a committing peer. Endorsing peers do dedicate blocks, but committing peers don't endorse transactions.

Endorsing peers verify the smart contracts, and perform a chaincode feature to imitate the transaction. The output is actually the chaincode benefits, a pair of key/value designs which were read through in the chaincode, as well as the set of keys/values which were composed by the chaincode. The proposal reply gets delivered returned to the client, together having an endorsement signature. These proposal responses are actually delivered to the orderer being purchased. The orderer then buys the transactions to a block, that it forwards to the committing as well as endorsing peers. The RW sets are actually used to confirm that the transactions continue to be valid ahead of when the information in the ledger as

well as planet express is actually kept up to date. Lastly, the peers asynchronously notify the smart contracts of the success or maybe failure of the transaction.

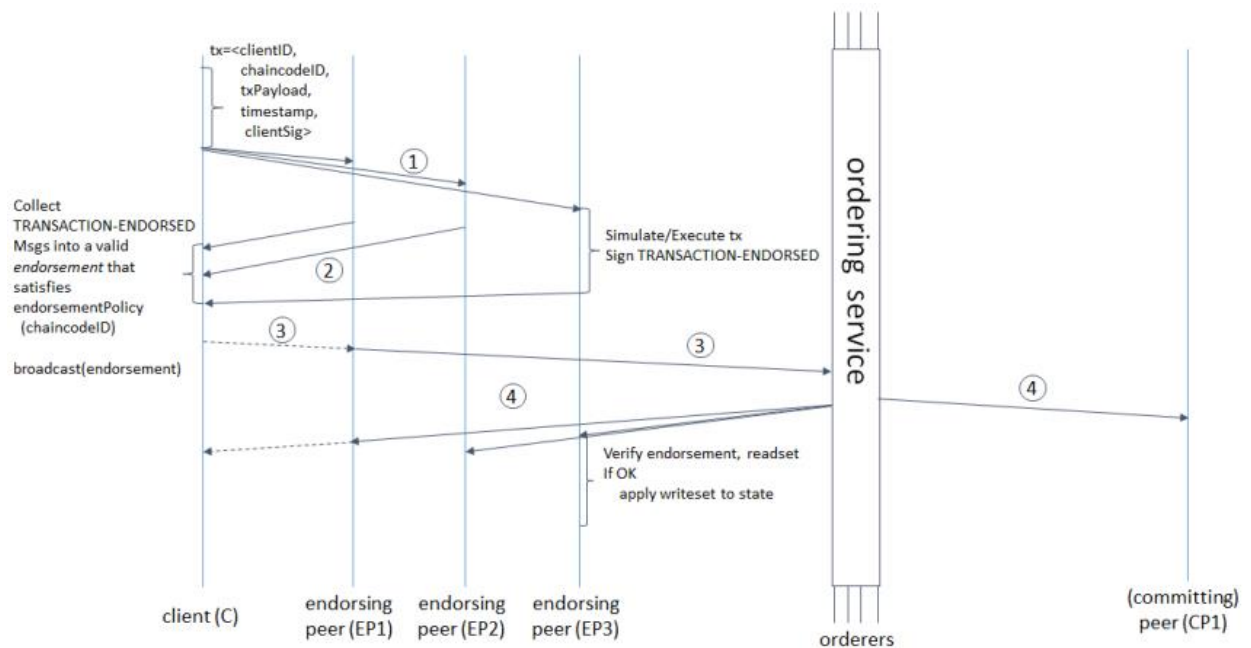


Figure 2.6-Illustration of one possible transaction flow (common-case path)

In the Blockchain network, the shared ledger must be entered in order. Transactions must be conducted to ensure that the world's states are effective when there is a commitment to the network. Unlike Bitcoin blockchains, which are sorted by decryption password pickup or mining, Hyperledger allows opting out of the network to select an operating system that runs the network, which is the most suitable operating system for the network. This module is very beneficial to Hyperledger Fabric enterprise applications.

It's crucial that you be aware that the state of the network is actually maintained by peers, and not by the ordering service or maybe the client. Usually, you are going to design the program of yours so that various devices in the network play various roles. That's, devices which are a part of the buying service shouldn't be set up to also endorse or even commit transactions, and the other way round. Nevertheless, there's an overlap between endorsing as well as committing peers on the product. Endorsing peers should get access to and hold sensible contracts, additionally to fulfilling the job of a committing peer. Endorsing peers do commit blocks, but committing peers don't endorse transactions [21].

Endorsing peers verify the client signature, and perform a chaincode feature to simulate the transaction. The output is actually the chaincode benefits, a set of key/value designs which were read in the chaincode (Read set), as well as the set of keys/values which were composed by the chaincode. The proposal reply gets sent back to the client, together with an endorsement signature. These proposal responses are actually delivered to the orderer to be purchased. The orderer then buys the transactions to a block, which it forwards to the committing as well as endorsing peers. The RW sets are actually used to confirm that the transactions continue to be valid before the content of the ledger as well as planet express is

actually kept up to date. Lastly, the peers asynchronously notify the client program of the success or maybe failure of the transaction.

But totally no one will believe, sooner or later even the most convinced anarchist should make a decision about trust in another individual. In the case of crypto-currencies, as the first practical use of blocking technology, such trust is based on the consensus of the vast majority of users regarding the financial transactions that have occurred.

It should be clarified that consent is required not from ordinary users of crypto currency or, in general, information systems using blocking technology, but among the miners who specialize in forming the next blocks of the chain and holders of a complete copy of the blockage. As the chain of blocks grows, the relative number of subjects that determine consensus decreases. The process of forming and accepting new blocks from a peer model tends to be centralized. This potentially creates a situation where the trust of ordinary users to a certain crypto currency or information system can be greatly shaken.

At the initial stage of the development of systems using block libraries, for example Bitcoin, anyone interested in entering the consensus could join the number of persons participating in the development of the consensus. The rule has not changed today, but the barrier to full participation in the process has gradually grown. This marked increase in the size of the blockade itself and the growth in the requirements for computing power.

The Hyperledger architecture provides three consecutive systems: SOLO, Kafka, and Simplified Byzantine Fault Tolerance (SBFT), which have not yet been implemented in v1.0.

SOLO is a ordering utility that is commonly used by developers which are experimenting with the Hyperledger fabric network. SOLO consists of a single sequence node.

Kafka is a Hyperledger Fabric ordering mechanism that is recommended for production use. The command system uses Apache Kafka, an open source stream processing platform that provides a unified, high-pass, low-frequency platform for real-time data feed processing. In this case, the data contains approved transactions sets. Kafka provides a fault tolerant solution for system classification [26].

SBFT is a Simplified Byzantine Fault Tolerance. This sequential mechanism is a collision-tolerant and Byzantine fault-tolerant, so an agreement can be reached in the presence of a malicious node or a defective node. The Hyperledger Structure Community has not yet implemented this system, but it is planning.

These three ordering mechanisms are alternative ways of indicating acceptance of a transactions.

A system or chain of blocks is an ordered database that stores all information on all computers interacting with the network. The database keeps a list of signed blocks with a timestamp, a link to the previous block and other necessary data. Production never stops, and the list of blocks is constantly replenished.

Blockchain technologies and projects are a convenient and promising system, but they do not exist for all types of crypto currency (there are a lot of them and constantly becomes more and more), but for example, bitcoin exists.

Because many users are now interested in how to make a blockchain project, with how to start, etc.

Technology blockchain and why is it needed? A close analogy is the patient's medical history. It records entries with a note of the date and time, but records in hindsight are not possible. Keys for access to information blocks - records are available only to the doctor and the patient. Access to all blocks in history can only be obtained by those to whom the doctor or patient will provide their keys.

2.4 Network structure and architecture of application

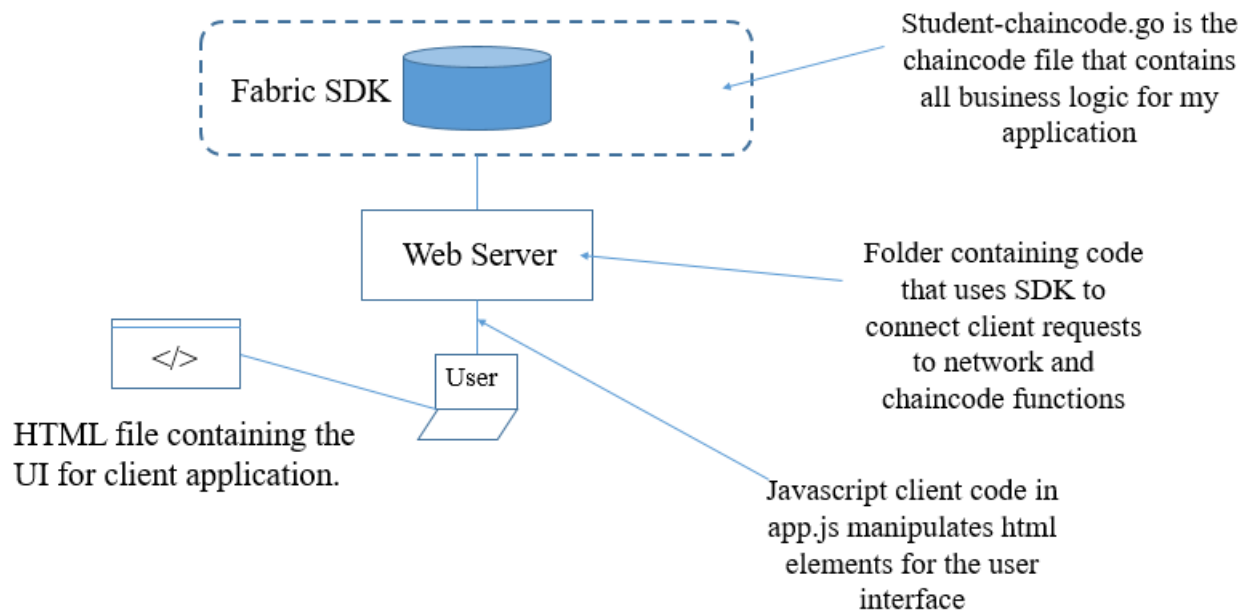


Figure 2.7 – Architecture of developed application

Let's look closer to the each element of the system.

The main part of my application is SDKs, which include on itself Node SDK, Java SDK and CLI, so it has been combined all notions and called them Fabric SDK. Being a reminder, smart contracts are actually computer programs that have logic to perform transactions and change the state of the assets stored to the ledger. Hyperledger Fabric smart contracts are actually known as chaincode and are actually written in Go. The chaincode serves when the online business logic for a Hyperledger Fabric network, in that the chaincode directs the way you manipulate assets to the community.

Student-chaincode.go is the chaincode file that contains all my application logic. This is deployed on the network peers and from my application's backend it was used the SDK to call functions within smart contract.

```
func (s *SmartContract) recordTuna(APIStub shim.ChaincodeStubInterface, args []string) sc.Response {
    if len(args) != 5 {
        return shim.Error("Incorrect number of arguments. Expecting 5")
    }

    var tuna = Tuna{ Vessel: args[1], Location: args[2], Timestamp: args[3], Holder: args[4] }

    tunaAsBytes, _ := json.Marshal(tuna)
    err := APIStub.PutState(args[0], tunaAsBytes)
    if err != nil {
        return shim.Error(fmt.Sprintf("Failed to record tuna catch: %s", args[0]))
    }

    return shim.Success(nil)
}
```

Figure 2.8 – recordTuna() function from tuna-chaincode.go file

Chaincode in fabric is nothing else than the usual smart contract. The chaincode can be written on Go, Java and, in the near future, JavaScript. The article deals with the implementation of a chaincode on Go, but I think this approach is suitable for Java implementations.

A little foreword about how the chaincode is used on the side of the node:

1. Node receives a request for a de-job and looks for a docker-container. If it does not find it, creates a new docker-container with a chaincode inside, if it finds it, it uses it.

2. After creating the container, it starts the executable file of the chaincode, which connects to gRPC to the node and starts listening to its instructions.

3. When an Invoke / Query request arrives on the node, it simply sends the commands to the executing files to the executing files, they process them and send the result back.

Accordingly, initially we were looking for a way to deceive the node, so that it would not look for a container, but this did not succeed. Quick reading of docker-compose files revealed an interesting parameter - `core_chaincode_mode`, with which you can force the node not to create or search for containers, but simply trust the environment.

In the go language, there are no concepts such as class, constructor, and destructor (along with the corresponding reserved words). However, there are structures borrowed from C++, to which you can bind functions, so in Go you can create code in the style of OOP. The presence of a "garbage collector" makes it easier to work with memory, compared to C or C ++. There are also pointers, but arithmetic is not provided for them. Therefore, even knowing the address of a variable, it is impossible to move relative to it. This is done for security reasons. There are also no header files and implicit type conversions. Multithreading is supported at the level of the language, channels are used to link the streams [29].

To begin interfacing with Hyperledger Fabric clients use HTML file called `index.html`, which contains the necessary UI for client application. Here are the main functions that a user can use to create and store a record. After the user fills in all the suggested fields and presses the button “Create” or “Query”, the entry is

sent for verification and confirmation to other nodes, after which it is successfully saved to the block, and the network state for all nodes is updated.

```

<div class="form-group">
  <label>Create Mark Record</label>
  <h5 style="color:green;margin-bottom:2%" id="success_create">Success! Tx ID: {{create_student}}</h5>
  <br>
  Enter mark id: <input class="form-control" type="text" placeholder="Ex: 11" ng-model="student.id">
  Enter Student name: <input class="form-control" type="text" placeholder="Ex: Kirill Sosnin" ng-model="tuna.vessel">
  Enter discipline: <input class="form-control" type="text" placeholder="Ex: Packet Switching and Softswitching Networks" ng-model="student.holder">
  Enter mark: <input id="createName" class="form-control" type="text" placeholder="Ex: 97" ng-model="student.longitude">
  Enter timestamp: <input class="form-control" type="text" placeholder="Ex: 13.09.2018" ng-model="student.timestamp">

  <input id="createSubmit" type="submit" value="Create" class="btn btn-primary" ng-click="recordStudent()">
</div>

```

Figure 2.9 - Creating a record in the client part

But simply filling in the fields is not enough, there must be a program that will process the entered transactions and send them for verification to other peers. In our case, this action is answered by the app. The file contains all the necessary logic in order for incoming transactions to be correctly processed by peers. JavaScript is not designed to create standalone applications. The JavaScript program is built directly into the source of the HTML document and interpreted by the browser as the document loads. Using JavaScript, you can dynamically change the text of a loaded HTML document and respond to events related to visitor actions or changes in the state of a document or window. An important feature of JavaScript is object orientation. A programmer can access numerous objects, such as documents, hyperlinks, forms, frames, etc. Objects are characterized by descriptive information (properties) and possible actions (methods). Since JavaScript provides user interaction with a Web page after downloading it, developers usually use it to solve the following tasks:

- dynamic addition, editing and deletion of HTML elements and their values;
- check the contents of web-forms before sending to the server;
- creation of cookies on the client's computer for saving and retrieving data on subsequent visits.

HTML was developed as a language for the exchange of technical and scientific documentation, appropriate for use by those who are not specialists in format. HTML effectively coped with the intricacy issue by defining a little range of semantic and structural components utilized to produce simple and easy but superbly created files. Apart from simplifying the framework of the document, HTML supports hypertext. Multimedia functions had been added later on. At first, the HTML language was conceived as well as developed as a means of structuring and formatting documents without the binding of theirs to the ways of

reproduction. Preferably, text with HTML markup must be reproduced on hardware with different specialized equipment (color display of a contemporary computer, monochrome display of the organizer, a small display of a cell phone or maybe applications and equipment for voice playback of copy) without structural and stylistic distortions [29].

```
$scope.queryStudent = function(){
    var id = $scope.student_id;
    appFactory.queryStudent(id, function(data){
        $scope.query_student = data;
        if ($scope.query_student == "Could not locate the mark"){
            console.log()
            $("#error_query").show();
        } else{
            $("#error_query").hide();
        }
    });
}

$scope.recordStudent = function(){
    appFactory.recordStudent($scope.student, function(data){
        $scope.create_student = data;
        $("#success_create").show();
    });
}
```

Figure 2.10 – Functions queryStudent() and recordStudent() in app.js

The meaning of the work of the blockchain network is to control the integrity of the data and counteract their falsification. In the most famous blockchain - in Bitcoin's blockchain - all the nodes compete for the right to verify the transaction, because the winner receives a reward in the form of bitcoins. In the HLF, as we have already said, network members have different rights and perform different tasks. Therefore, a separate service is responsible for the formation of the units.

The service work is one of the most interesting technical details in the HLF. Unlike other detachments, different mechanisms for reaching consensus are possible here, which allow us to form the next block. The goal of the Hyperledger project is to build an open source framework for creating and executing robust applications and industry platforms for conducting business transactions. The technical community built around the Hyperledger project is an excellent platform for sharing knowledge and suggestions between users and developers, which has a positive impact on the development of the project. In one of the implementations of blocking technology within the framework of Hyperledger-Fabric, we use different proof-of-work transaction confirmation algorithms that demonstrate good throughput. This all makes Hyperledger an excellent contender for the role of a block system for business.

The last element of the system is the part responsible for communication of client application with a distributed database. To do this, Hyperledger has a whole directory that stores a set of programs that store data entered by the user into the block, and provide information for queries.

```
var mark id = array[0]
var timestamp = array[2]
var student name = array[1]
var mark = array[4]
var student name = array[3]

var fabric_client = new Fabric_Client();

// setup the fabric network
var channel = fabric_client.newChannel('mychannel');
var peer = fabric_client.newPeer('grpc://localhost:7051');
channel.addPeer(peer);
var order = fabric_client.newOrderer('grpc://localhost:7050')
channel.addOrderer(order);
```

Figure 2.11 – Set up the fabric network in recordStudent() function

New block-services in the IBM Cloud: this type of services help to create and manage block-network networks to ensure the operation of a new class of distributed financial applications. Now, developers will be able to create digital assets and relevant business logic, which will open up the possibility for a more secure and confidential transfer of assets between participants in the test network of blockers with limited and protected access rights. Using new block technologies from IBM, available on GitHub, developers can use fully integrated DevOps tools to create, deploy, launch and monitor blocking applications on the IBM Cloud. Blocker applications can access existing transactions on distributed servers through the API, thus supporting new types of payment, transaction, supply and business procedures. As a result of the project to open the source code of block-technology, the business received an improved identity management process, which is based on the latest achievements in cryptography, monitoring and control of privacy and confidentiality, as well as smart contracts based on Java and Go with the ability to determine the rights of users who have access to these contracts.

2.5 Process of creating an application for the recording of students' progress

In order to successfully install Hyperledger Fabric, you should have the following features installed on your computer: cURL, Node.js, npm package manager, Go language, Docker, and Docker Compose.

Next, we will download the latest released Docker images for Hyperledger Fabric, and tag them with the latest tag. To confirm and see the list of Docker images it has just downloaded, run:

```
$ docker images
```


rill-VirtualBox: ~			
hyperledger/sawtooth-rest_api			
latest	b3314238539e	4 weeks ago	177MB
hyperledger/sawtooth-tx_intkey_python			
latest	80d5e35f93e4	4 weeks ago	171MB
hyperledger/fabric-ca			
latest	72617b4fa9b4	2 months ago	299MB
hyperledger/fabric-ca			
x86_64-1.1.0	72617b4fa9b4	2 months ago	299MB
hyperledger/fabric-tools			
latest	b7bfddf508bc	2 months ago	1.46GB
hyperledger/fabric-tools			
x86_64-1.1.0	b7bfddf508bc	2 months ago	1.46GB
hyperledger/fabric-orderer			
latest	ce0c810df36a	2 months ago	180MB
hyperledger/fabric-orderer			
x86_64-1.1.0	ce0c810df36a	2 months ago	180MB
hyperledger/fabric-peer			
latest	b023f9be0771	2 months ago	187MB
hyperledger/fabric-peer			
x86_64-1.1.0	b023f9be0771	2 months ago	187MB
hyperledger/fabric-javaenv			
latest	82098abb1a17	2 months ago	1.52GB
hyperledger/fabric-javaenv			
x86_64-1.1.0	82098abb1a17	2 months ago	1.52GB

Figure 2.12 – Hyperledger Fabric docker images

To install the Hyperledger Fabric sample code which will be used in project, do:

```
$ git clone https://github.com/hyperledger/fabric-samples.git
```

Now that we have successfully installed Hyperledger Fabric, we can walk through setting up a simple network that has two members. To refer back to our demonstrated scenario, the network includes asset management of each transaction verified, transferred, each maintaining two peers and an ordering service.

We will use Docker images to bootstrap our first Hyperledger Fabric network. It will also launch a container to run a scripted execution that will join peers to a channel, deploy, and instantiate the chaincode, and execute transactions against the chaincode.

Next, execute command:

```
$ ./byfn.sh -m up
```

The Docker is actually an open platform for the improvement, operation as well as delivery of uses. Docker is created to rapidly deploy the uses of yours. To us the docker, you are able to separate the application of yours from the infrastructure of yours and deal with the infrastructure as being a managed program. Docker makes it possible to distribute the code of yours more quickly, more quickly testing, faster spreadsheet programs as well as a shorter time between writing code and operating code. Docker performs this with a light-weight container virtualization platform, utilizing procedures as well as utilities which help control as well as deploy the uses of yours.

From the core of its, the docker enables you to operate almost any program which is properly isolated in the container. Protected isolation enables you to run numerous containers on the identical host simultaneously. The little dynamics of the container, which runs with no extra hypervisor load, enables you to get much more from the iron of yours.


```

kirill@kirill-VirtualBox:~/fabric-samples/first-network$ . byfn.sh -n up
Generating certs and genesis block for with channel 'mychannel' and CLI timeout of '10'
Continue (y/n)? y
proceeding ...
/home/kirill/fabric-samples/first-network/./bin/cryptogen

#####
#### Generate certificates using cryptogen tool #####
#####
org1.example.com
org2.example.com

/home/kirill/fabric-samples/first-network/./bin/configtxgen
#####
##### Generating Orderer Genesis block #####
#####
2018-06-03 23:07:35.439 +06 [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-06-03 23:07:35.457 +06 [common/tools/configtxgen] doOutputBlock -> INFO 002 Generating genesis block
2018-06-03 23:07:35.460 +06 [common/tools/configtxgen] doOutputBlock -> INFO 003 Writing genesis block

#####
### Generating channel configuration transaction 'channel.tx' ###
#####
2018-06-03 23:07:35.516 +06 [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-06-03 23:07:35.542 +06 [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 002 Generating new
channel configtx
2018-06-03 23:07:35.601 +06 [common/tools/configtxgen] doOutputChannelCreateTx -> INFO 003 Writing new ch
annel tx

#####
##### Generating anchor peer update for Org1MSP #####
#####
2018-06-03 23:07:35.653 +06 [common/tools/configtxgen] main -> INFO 001 Loading configuration
2018-06-03 23:07:35.681 +06 [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 002 Generating a
nchor peer update
2018-06-03 23:07:35.693 +06 [common/tools/configtxgen] doOutputAnchorPeersUpdate -> INFO 003 Writing anch
or peer update

```

Figure 2.13 – Hyperledger Fabric sample network

In a blockchain program, the blockchain is going to store the state of the database, in addition to the immutable history of transactions the produced that state. A client application is going to be utilized to send out transactions to the blockchain. The smart contracts are going to encode some (in case not all) of the company reason. Apps make use of APIs to run smart contracts. Inside Hyperledger Fabric, these smart contracts are labeled as chaincode. These contracts are actually hosted on the network, as well as identified by edition and name. APIs are actually accessible with a software program development package, or maybe SDK. Currently, Hyperledger Fabric has 3 choices for developers: Node.js SDK, Java SDK, and CLI [21].

After we were convinced that the test network is functional and all its components are actually added, you can proceed to create an application for accounting students' progress. Move to the directory containing the basis of our application and run it using the following commands:

```

$ cd education/LFS171x/fabric-material/tuna-app
$ ./startFabric.sh

```

Node or perhaps Node.js is actually a program platform depending on the V8 engine (translating JavaScript into machine code), which turns JavaScript from a very specialized language into a general purpose language.

Node.js gives the capability of JavaScript to work together with I/O products through the API of its (composed in C), connect different outside libraries created in languages that are different, supplying calls to them from JavaScript code. Node.js is utilized largely on the server. Node.js is based on asynchronous and event-oriented (or maybe reactive) programming with non blocking I / O.

```

kirill@kirill-VirtualBox: ~/education/LFS171x/fabric-material/tuna-app
fabric-samples package-lock.json registerUser.js src
kirill@kirill-VirtualBox:~/education/LFS171x/fabric-material/tuna-app$ ./startFabric.sh

# don't rewrite paths for Windows Git Bash users
export MSYS_NO_PATHCONV=1

docker-compose -f docker-compose.yml down
Removing cli ... done
Removing peer0.org1.example.com ... done
Removing ca.example.com ... done
Removing orderer.example.com ... done
Removing couchdb ... done
Removing network net_basic

docker-compose -f docker-compose.yml up -d ca.example.com orderer.example.com peer0.org1.example.com couc
hdb
Creating orderer.example.com ... done
Creating peer0.org1.example.com ... done
Creating orderer.example.com ...
Creating couchdb ...
Creating peer0.org1.example.com ...

# wait for Hyperledger Fabric to start
# incase of errors when running later commands, issue export FABRIC_START_TIMEOUT=<larger number>
export FABRIC_START_TIMEOUT=10
# echo ${FABRIC_START_TIMEOUT}
sleep ${FABRIC_START_TIMEOUT}

# Create the channel
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/nsp/users/Admi
n@org1.example.com/nsp" peer0.org1.example.com peer channel create -o orderer.example.com:7050 -c mychann
el -f /etc/hyperledger/configtx/channel.tx
2018-06-04 07:52:01.391 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections init
ialized

```

Figure 2.14 – Hyperledger Fabric sample network

After that, we should register Admin and User components of our application and activate Server part. To do this I executed next commands:

\$ node registerAdmin.js

```

kirill@kirill-VirtualBox:~/education/LFS171x/fabric-material/tuna-app$ node registerAdmin.js
Store path:/home/kirill/.hfc-key-store
Successfully enrolled admin user "admin"
Assigned the admin user to the fabric client ::{"name":"admin","mspid":"Org1MSP","roles":null,"affiliatio
n":"","enrollmentSecret":"","enrollment":{"signingIdentity":"537f155fbc41d579e3bb4af0b82512c1b891d7914b2e
30dede16b4cbbc3485f8","identity":{"certificate":"-----BEGIN CERTIFICATE-----\nMIICAjCCAAIgaWIBAgIUJ5rHpog
Eq0CEk2N/jCf001Hn/+UwCgYIKoZIzj0EAwIw\nczELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbgGmb3JuaWExFjAUBgNVBAcTDVNH\n
biBGcmFuY2l2Y28xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTNE2NhLm9yZzEuZXhhbXBsZS5jb20wHhcNMjgwNjA
0MTAyMjAwHhcNMjgwNjA0MTAyMjAwHhQ8wDQYDVQQLEwZjbGllbnQxZjAMBgNVBAcTbWZkbWUwFkEwYHkoZi\nnzj0CAQYIKoZI
zj0DAQcDQgAE8oPT0AaTlh+OXNvFnN323XJU00MdxMMHjjpLlmu\nnbk3QKfgAIdFju9q9GJa7UXOdqM8IQHZVhrXz5apXqICR16NsMGo
wDgYDVVR0PAQH/\nBAQDAgeAMAwGA1UdEwEB/wQCMAAwHQYDVRO0BBYEFM1vN9XVbGEGHvJ0LAIBYblj\nnwjZJMCsGA1UdIwQkMCAIEIS
qg3NdtruuLoM2nAYUDFFBNMarRst3dusalc2Xk18\nnMAoGCCqGSM49BAMCA0gAMEUCIQDvdazRr+LhVqKn2j4nFUBwq+qk1qT2hZbvs5B
P\nXI2ngaIgvDHxzf69pRCEPNT/cJob2xb2bDNML9RJXnG8rBPUPVY=\n-----END CERTIFICATE-----\n"}}}
kirill@kirill-VirtualBox:~/education/LFS171x/fabric-material/tuna-app$ █

```

Figure 2.15 – Terminal window for admin registration

Then, it should be registered the person with admin root. It can be done by typing terminal command:

\$ node registerUser.js

```
kirill@kirill-VirtualBox:~/education/LFS171x/fabric-material/tuna-app$ node registerUser.js
Store path:/home/kirill/.hfc-key-store
Successfully loaded admin from persistence
Successfully registered user1 - secret:csBwTgWsYRqE
Successfully enrolled member user "user1"
User1 was successfully registered and enrolled and is ready to interact with the fabric network
kirill@kirill-VirtualBox:~/education/LFS171x/fabric-material/tuna-app$
```

Figure 2.16 – Terminal window for User registration

And finally, I activated the server part of my decentralized application to interact with it by command:

\$ node server.js

```
kirill@kirill-VirtualBox:~/education/LFS171x/fabric-material/tuna-app$ node server.js
Live on port: 8000
```

Figure 2.17 – Terminal window for server part

This command will start the network. As on the picture above are shown my application now is in working process, so I can easily interact with it using any Web browser just by typing on the address bar “localhost:8000”.

Then, on a browsers’ window you should obtain user interface, looks similar with picture below:

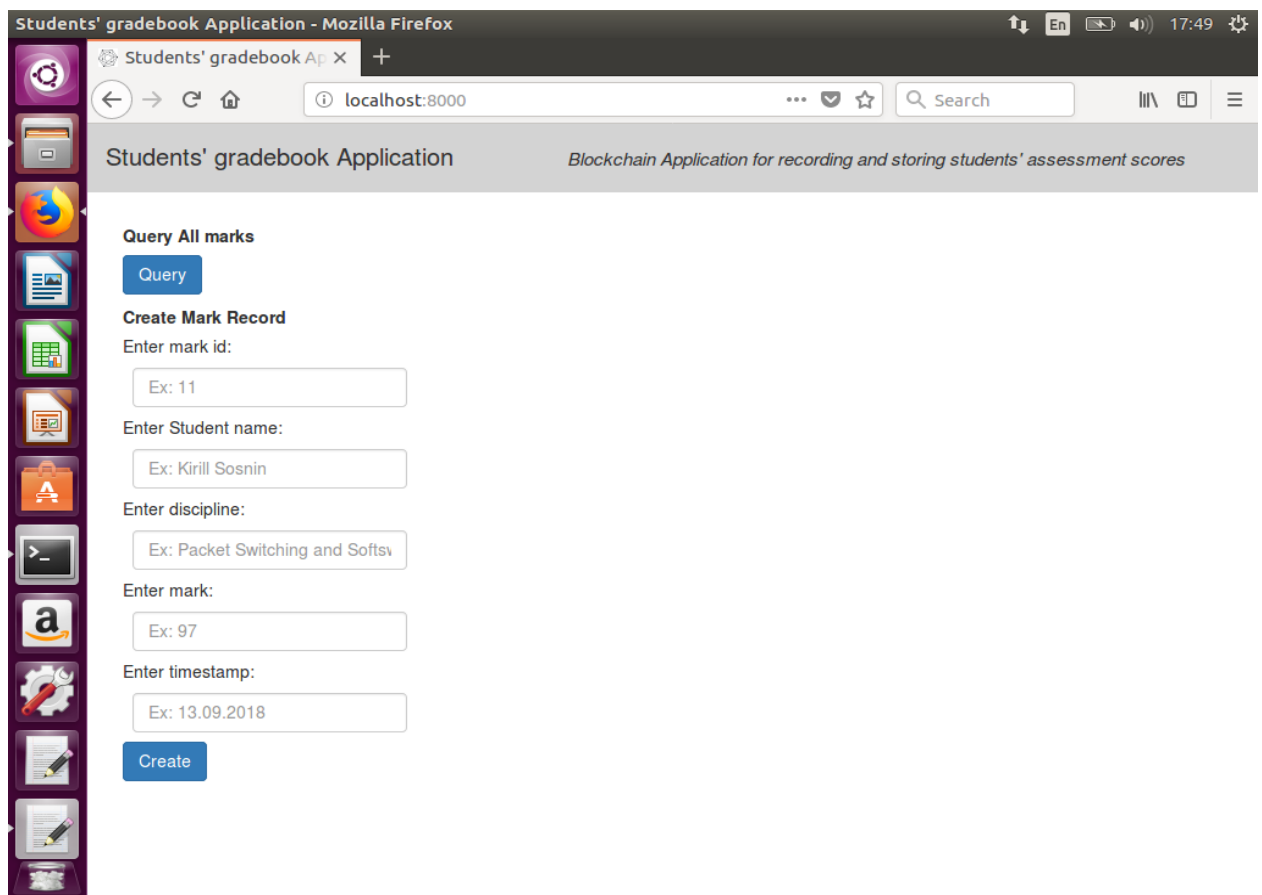


Figure 2.18 – User interface of application

As you can see my application consist of 2 main parts:

1) User can find out all marks, already stored in a distributed ledger, by clicking the button “Query”. This option is currently unavailable, it is in developing process now.

2) Creations of a new records on the special fields and confirming this records by pressing the button “Create”.

Created in this diploma project blockchain application gives to User opportunity to fill such fields as:

- student name;
- discipline;
- mark;
- timestamp.

Create Mark Record

Enter mark id:

Enter Student name:

Enter discipline:

Enter mark:

Enter timestamp:

Create

Figure 2.19 – Fields for creating the record of mark

After filling all fields User can create the necessary record by pressing button “Create”. After that, record will be accepted by network and stored on distributed database. So the current state of blockchain will change. You can see these changes in a terminal window.

The property of immutability is actually attained from the usage of cryptography, and never at that cost of trust in the business or even individuals. 2 of probably the simplest cryptographic algorithms employed at a block structure are actually electric signatures and hash features which make sure the integrity of transactions and therefore are accountable for authorization. A crucial characteristic of the transaction log within the blockroom is the immutability of its. This particular property would mean that you are able to not silently delete a transaction in the log or even include a brand new one to the middle of its.

Create Mark Record

Success! Tx ID:
f4270152dcf16d28283318bdc9da97db41f5aee97a044c1cc40dd9f794836dae

Enter mark id:

Enter Student name:

Enter discipline:

Enter mark:

Enter timestamp:

Figure 2.20 – Adoption of a new record

```
submit recording of a tuna catch:
[ '11', '97', '13.09.2018', 'PS&SN', 'Kirill Sosnin' ]
Store path:/home/kirill/.hfc-key-store
Successfully loaded user1 from persistence
Assigning transaction_id: f4270152dcf16d28283318bdc9da97db41f5aee97a044c1cc40dd9f794836dae
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - "OK"
info: [EventHub.js]: _connect - options {}
The transaction has been committed on peer localhost:7053
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

Figure 2.21 – Successful ledger state change

So far, there has been covered the components of Hyperledger Fabric's framework, including the different types of nodes in the network, private channels, and database features. It has also installed and spun up our very own test network, deep dived into chaincodes smart contract programming, and gone through a demonstrated example, detailing how Fabric is so unique. So, now, it has gotten to the really exciting part, where I combine all of these concepts into a working sample application.

We'll see how a user can interact with a network through an application that enables users to query and update a ledger. My application handles user interface and submits transactions to the network, which then call chaincodes. Fabric currently has three options for developers: a Node SDK, Java SDK and a command line interface or CLI [21].

In a nutshell, reading or writing the ledger is known as a proposal. This proposal is built by Fabric application via the SDK, and then sent to a blockchain peer. The peer will communicate to its application-specific chaincode container.

The chaincode will run the transaction. If there are no issues, it will endorse the transaction and send it back to our application. My application, via the SDK, will then send the endorsed proposal to the ordering service. The order will package many proposals from the whole network into a block, which is then broadcast to the peers in the network. Finally, the peer will validate the block and write it to its ledger. The transaction is now live. By the end, familiar with how to use the Node.js SDK to interact with the network, and, therefore, a ledger, and understand how an application chaincode network and ledger all interact with one another.

2.6 Calculation of main network parameters

Suppose, that we have a blockchain network, consist of 6 nodes, where the node-graph looks as in Figure 2.

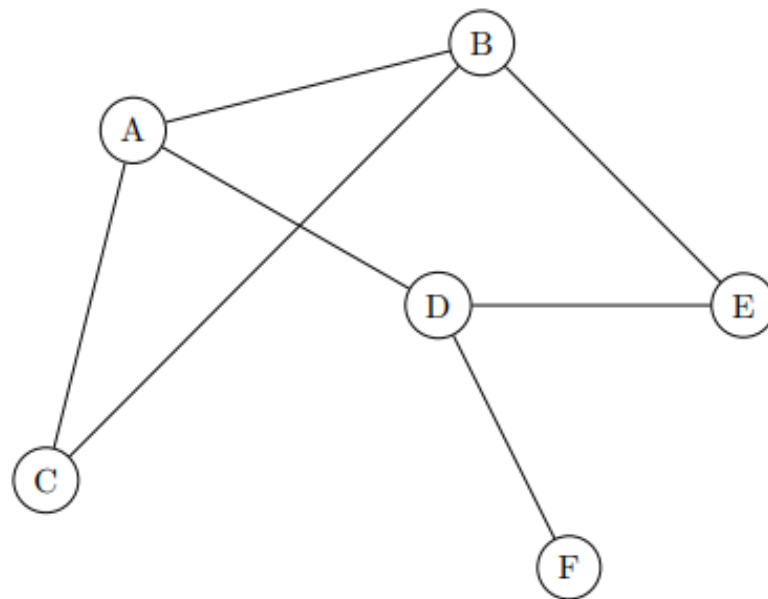


Figure 2.22. The node network

The given variables (as the name suggests) from Figure 1 have to be given. There is no way the creator of the blockchain can change these. Many of them are dictated by the users/nodes of the blockchain causing them to change regularly. This example will work with the following:

Headersize – 72 bytes. [12]

Transaction size – 102 400 bytes. [12]

№ of nodes in the system – 6.

Blocksize – 1 Mb (1 048 576 bytes).

Difficulty cryptopuzzle = 500 [12]

Table 2.1 - Mining power

Mining power per node(hashes/second)	A	B	C	D	E	F
	1000	1500	500	750	900	1100

Table 2.2 - Bandwidth of nodes [4]

Bandwidth		A	B	C	D	E	F
	A	x	9	10	7	-	-
	B	-	x	6	-	10	-
	C	-	-	x	-	-	-
	D	-	-	-	x	7	10
	E	-	-	-	-	x	-
	F	-	-	-	-	-	x

Total mining power = $\sum_{i \in \text{nodes}} m(i) = 1000 + 1500 + 500 + 750 + 900 + 1100 = 5750$ hashes/second.

Number of transactions per block(2.3.1):

$$N = \frac{\text{blocksize} - \text{headersize}}{\text{transactionsiz}} \quad (2.1)$$

$$N = \frac{1\,048\,576 - 72}{102\,400} = 10.24$$

$$\text{Blockfrequency} = \frac{\text{total mining power}}{\text{difficulty cryptopuzzle}} \quad (2.2)$$

$$\text{Blockfrequency} = \frac{5750}{500} = 115 \text{ blocks per minute}$$

Let's calculate number of transactions n per second:

$$n = \frac{\text{Blockfrequency}}{60} * N \quad (2.3)$$

$$n = \frac{115}{60} * 10.24 = 19.63$$

The inter nodes time is a bit tricky to calculate. This is because one needs the bandwidth between all nodes, even those without a direct line. In order to get these, one first needs the 'quickest' route between the not directly linked nodes.

$$\text{Inter nodes times} = \frac{\text{blocksize}}{b(i,j)} \quad (2.4)$$

where

$b(i,j)$ – bandwidth between nodes i and j, $i \neq j$

$$AE = ABE = \frac{1}{9} + \frac{1}{10} = \frac{19}{90}, \text{sec}$$

$$BD = BED = \frac{1}{10} + \frac{1}{7} = \frac{17}{70}, \text{sec}$$

$$AF = ADF = \frac{1}{9} + \frac{1}{10} = \frac{19}{90}, sec$$

$$BF = BEDF = \frac{1}{10} + \frac{1}{7} + \frac{1}{10} = \frac{12}{35}, sec$$

$$CE = CAEB = \frac{1}{10} + \frac{1}{9} + \frac{1}{10} = \frac{14}{45}, sec$$

$$CF = CADF = \frac{1}{10} + \frac{1}{7} + \frac{1}{10} = \frac{12}{35}, sec$$

$$CD = CAD = \frac{1}{10} + \frac{1}{7} = \frac{17}{70}, sec$$

$$EF = EDF = \frac{1}{7} + \frac{1}{10} = \frac{17}{70}, sec$$

$$\text{System width} = \max\{\text{inter nodes times}\} = \frac{12}{35}, sec$$

A fork is developed when two miners find the shine at different times. Both units will be destroyed on the network, and all ministers will work on the block they received first. Until the onte block is turned on, they are the 2 'last blocks' and it has not yet been discovered, but one of these blocks will eventually be included in the chain and will be discarded.

To begin with, it is assumed that the number of blocks found in the system for a given time interval follows the Poisson process. This is not a very strange assumption. Let's look at it this way: there are many nodes (large n) and a very small probability of finding a block (small p), creating a more or less "stable" λ . In this case, represents the number of blocks found per unit of time λ [17].

Now up for some declaring:

Nt = the number of blocks found in the time $(0,t) \sim \text{Poisson}(0.1t)$

$\Delta t \geq \text{system width}$

$P(\text{fork}) = P(\{\text{more than 1 block found in the time interval } (0, \Delta t)\})$

$= P(N(\Delta t) > 1)$

Take the complementary probability:

$$\begin{aligned} P(\text{fork}) &= 1 - P(N(\Delta t) \leq 1) \\ &= 1 - P(N(\Delta t) = 0) - P(N(\Delta t) = 1) \\ &= 1 - \frac{e^{-0.1 \cdot \frac{12}{35}} \left(0.1 \cdot \frac{12}{35}\right)^0}{0!} - \frac{e^{-0.1 \cdot \frac{12}{35}} \left(0.1 \cdot \frac{12}{35}\right)^1}{1!} \\ &= 1 - e^{-0.1 \cdot \frac{12}{35}} - 0.1 \cdot \frac{12}{35} \cdot e^{-0.1 \cdot \frac{12}{35}} \end{aligned}$$

$$\approx 0.966295 - 0.03313013$$

$$\approx 0.00057449$$

In other words the probability of a fork with these given parameters would be about 0.057%

2.6.1 Calculation of fault tolerance parameters for a network based on blockchain technology consisting of 2 nodes

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.

MTTR (Mean Time To Repair) is the average time to recover [30].

MTBF (Mean Time Between Failure) is a technical parameter that characterizes the reliability of a restored device, device or technical system. Measured statistically, by testing a variety of devices.

The probability of component failure during MTBF is 1, and if MTBF is measured in years, then the probability of failure of the component within one year is:

$$P = \frac{1}{MTBF} \quad (2.5)$$

The probability of failure of the entire node during the year is calculated as the sum of the probability of failure of individual components:

$$P_t = \sum P \quad (2.6)$$

Failure of a duplicated component will result in network failure only on condition that the backup component also fails during the time required to "hot" replace the component that failed first. If the guaranteed replacement time of the component is 24 hours (1/365 years), then the probability of such an event during the year:

$$P_d = \frac{P * P}{365} * 2 \quad (2.7)$$

The probability of simultaneous occurrence of these events is equal to the product of their probabilities.

For case, when component #2 first fails, and then component # 1, the probability will be the same.

Now, knowing the probability P_i of failure of each of the N components (duplicated and non-duplicated) of the server, you can calculate the probability of server failure within one year.

Let's perform the calculation as follows

As already mentioned the output of their system of any component will mean the failure of the server as a whole

Table 2.3 - Reliability parameters of network components [5]

Node components	Declared reliability			Number of components in the network	Probability of failure in view of duplication
	MTBF (hours)	MTBF (years)	Probability of refusal during the year		
Power Supply	135 000	15.41	0,06489	2	0,0000231
HDD	100 000	11.42	0,08757	2	0,0000420
Fan	80 000	9.13	0,10952	2	0,0000657
Motherboard	110 000	12.56	0,07962	2	0,0000347
Video card	90 000	10.27	0,09737	2	0,0000519
CPU	200 000	22.83	0,04380	2	0,0000105
For the network as a whole:			0,4828		0,0002279

Based on the above values, you can calculate the probability of node failure during the year:

$$P_{t(1)} = 0.4828$$

The probability of failure of the entire network during the year:

$$P_t = 0.0002279$$

The probability of simultaneous node failure is calculated by the formula:

$$P_{t(2)} = \frac{P_{t(1)} * P_{t(1)}}{365} * 2 = \frac{0.4828 * 0.4828}{365} * 2 = 0.001277$$

Since component failures are evenly distributed over time, knowing the probability of node failure within a year, you can determine the time it takes to fail (the time through which the node will fail with a 100% probability):

$$MTBF_c = \frac{1}{P_{t(2)}} = \frac{1}{0.001277} = 4388 \text{ years}$$

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

The availability factor is the probability that the computer system will be in any working state at any time.

This coefficient is determined by the formula:

$$K = \frac{MTBF_c}{MTBF_c + MTTR} = \frac{4388}{4388 + 24} = 0.99727$$

MTTR (Mean Time To Repair) — average recovery time.

2.7 Further development

In the course of this project, an application was made to record students' progress, however, many aspects are to be finalized/

Web applications benefit from the use of network effect due to the fact that they adhere to centralized storage of information. Built on common open protocols (eg TCP / IP and HTTP), Yelp, Facebook and Amazon benefit from the fact that all their users and, as a result, their data are in one place [8]. Thus, they not only gain an advantage over competitors who have less data, but also have complete control over how to monetize these data. As the functionality of distributed applications becomes more and more in demand, the cost of a fixed number of application coins grows as a function of supply and demand. In other words, if your slot machine is really memorable and begins to be in demand, then a limited number of tokens begins to grow in value. This is the principle underlying the ICO (initial coin offering), in which developers are engaged in collecting investments offering koin in exchange for capital to finance their projects.

The following tasks are to be implemented:

- organize permanent storage of data within the network;
- organize the distribution of data between nodes, in the project they are stored locally;
- add to the application role separation teacher / student.

3 Life activity safety section

3.1.General information on the labor protection of the enterprise

The aim of the project is to create distributed ledger software, which can save marks of students. Purpose of the work determines the field where project can be used. So one of the properly places is school. The project was created in the “STEP computer Academy”.

STEP Computer Academy is the largest international educational institution that specializes in IT education. Our work was developed in Almaty Branch of Academy.. Every work place has desktop computer or laptop. Students sit on two. My workplace is teacher' s table.

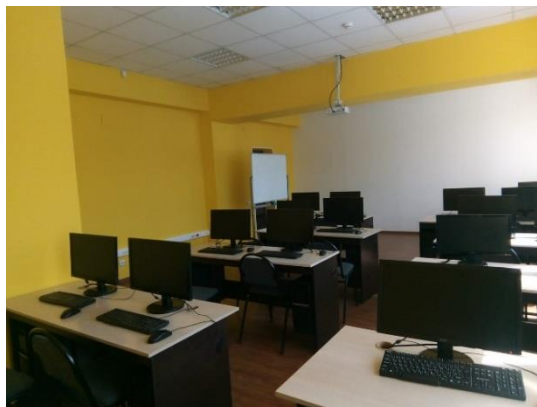


Figure 3.1 - Classroom

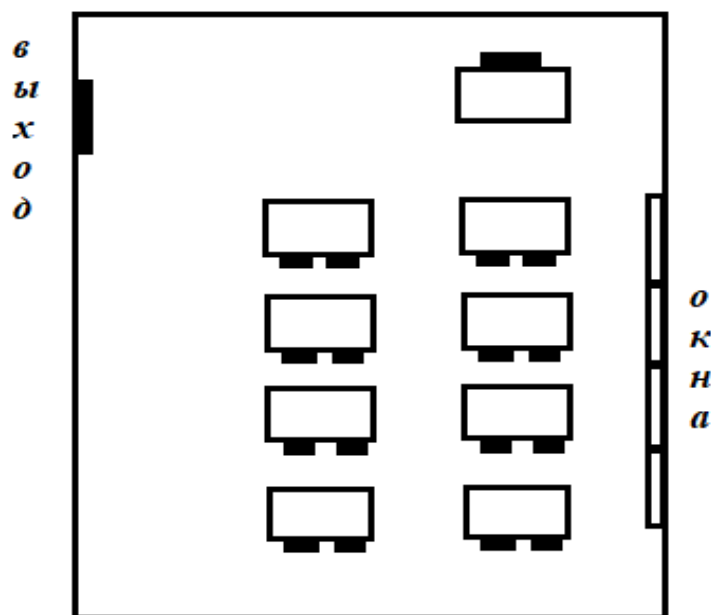


Figure 3.2 - Arrangement scheme of workplaces

Working conditions is the characteristics of the production process and the production environment, affecting the employee of the enterprise. It includes such parameters as nature of work (specialty, qualification, position), working hours, etc. The main point of working conditions is providing comfortable living conditions in production premises.

Working in Academy characterized by following conditions:

- working hours – 7;
- subject of work – developing application;
- position – intern.

Table 3.1 - Working equipment include desktop computer, projector, peripherals.

Device	Model	Amount
System unit	Neo Office (Ci5-6400 2,70Ghz/4GB/1000GB/HD510/ DVD-RW)	16
Monitor	HP 24es FHD IPS (T3M78AA) Silver	16
Projector	Epson G6450WU + ELPLM04 White	1
Laptop	HP Spectre BL x360	1
Laptop	Acer Aspire E572	1

During analysis of the work of the branch of "Computer Academy STEP-Almaty" were defined dangerous - harmful production factors.

There are the following types of hazardous and harmful production factors, accompanying the labor process at the enterprise [29]:

- physical;
- psychophysiological.

3.2 Measuring of noise and vibration level

Noise is a disorderly combination of a variety of sounds, therefore, to understand the physical basis of the formation and propagation of noise, its perception by man and influence on the organism, sound should be considered as an integral part of all noise, including the production noise.

The sources of noise and vibration in the auditoriums of the " STEP-Almaty Computer Academy" are:

- the roadway and cars are located in the immediate vicinity of the business center "Forte Leasing";
- working computers, servers, projectors and other devices.

In each of the branches of the " STEP-Almaty Computer Academy" there are nearly 15 to 20 computers that are working together and create an increased level of electromagnetic field intensity, which is also a factor that adversely affects the health of employees during long periods of working in these rooms.

Psychophysiological factors - changes in the state of health, physical and mental state of a person in the process of performing some kind of activity. Mental tension is a normal working condition that arises under the influence of labor activity. However, due to the action of certain features of the activity or the conditions in which it occurs, it can substantially increase. Such peculiarities are physiological discomfort, fear, lack of time, increased significance of erroneous actions, presence of interference, deficiency or excess of information, monotony (monotony of performed actions) or polytonia (the need for frequent switching of attention).

As a result of prolonged or systematic implementation of any actions in difficult conditions, there are harmful factors of a psychophysiological nature. These include, first of all, intellectual, sensory or physical overstrain, uncomfortable posture, chronic fatigue and stress, etc.

The effect of these factors leads to disruption of the mechanisms of human adaptation, development of endocrine disorders and neurotic conditions, diseases of the musculoskeletal system and sensory organs [29].

In addition, chronic fatigue in a number of occupations requiring increased attention can lead to emergencies.

Table 3.2 - The dangerous and harmful factors, that were found in the company

№	Name of hazardous and harmful production factor	Types of work, equipment, technological operations in which this production factor occurs
1	Physical	Working in the dark with switched on devices and opened windows
2	Psychophysiological	Doing monotone tasks during long period of time

When performing work on the personal computer noise level on a scale A should not exceed 50 dBA.

Vibration level in the room should not exceed admissible norms of vibration according to SNIP RK 3041-84 "Sanitary standards of vibration of workplaces" [3].

The rustling equipment (printers, etc.) which noise levels exceed rated, has to be out of rooms with the personal computer.

In rooms with operation of the personal computer temperature, relative humidity and speed of the movement of air on workplaces have to meet existing rules of a microclimate in Table 3.3.

Table 3.3 - Optimum norms of microclimate of rooms

Period of year	Work category	Air temperature, °C	Relative air humidity, %	Air movement speed, m/s
Cold	mild – 1a	22-24	40-60	0,1
Warm	mild – 1a	23-25	40-60	0,1

Optimum norms of a microclimate of rooms with the personal computer, C, %, m/s, in workplace GOST 12.1.005 – 88 "General sanitary and hygienic requirements to air of a working zone" [29].

The works performed belong to the category 1a sitting and not demanding the physical tension at which power consumption makes up to 120 kJ/h;

Content of harmful chemicals in air of production rooms with the personal computer should not exceed "Maximum permissible concentration of the polluting substances in atmospheric air of the inhabited places".

The workplace meets all GOST and SNIP on lighting, noise and vibrations, a room microclimate that does it optimum for work without deterioration in health. Workflow is also proceeding according to the requirements of SanPiN RK № 3073 and Art. 184. Sec. 19 of the Labor Code of the Republic of Kazakhstan, that is, satisfies the security requirements of jobs [18].

There were used air conditioning. What is air conditioning? This is device to maintain optimal climatic conditions in buildings, vehicles and other equipment. The type of conditioning that is used is Independent air conditioning systems. Autonomous air conditioning systems are supplied from the outside only with electric power, for example, cabinet air-conditioners and the like. Such conditioners have built-in compression refrigerating machines operating on freon - R22, R134A, R407C.



Figure 3.3 - Air Conditioning

3.3 Calculation of microclimate parameters

The following types of lighting are used in this building:

- natural – day. Natural lighting is actually the lighting effects of the areas with light coming out of the sky (direct or even reflected), penetrating over the gentle apertures in the external enclosing structures. It's split into lateral, top and combined;
- mixed - morning-day. Combined lighting effects is actually illumination, in which inadequate natural light is actually supplemented by man-made lighting;
- man-made - evening. Artificial lighting is actually the lighting of an area with energy sources of artificial light if the organic light lacks. It can easily be duty, security, emergency, and working. In case it's needed, component of the working or maybe emergency lighting lamps is actually utilized for duty lighting.

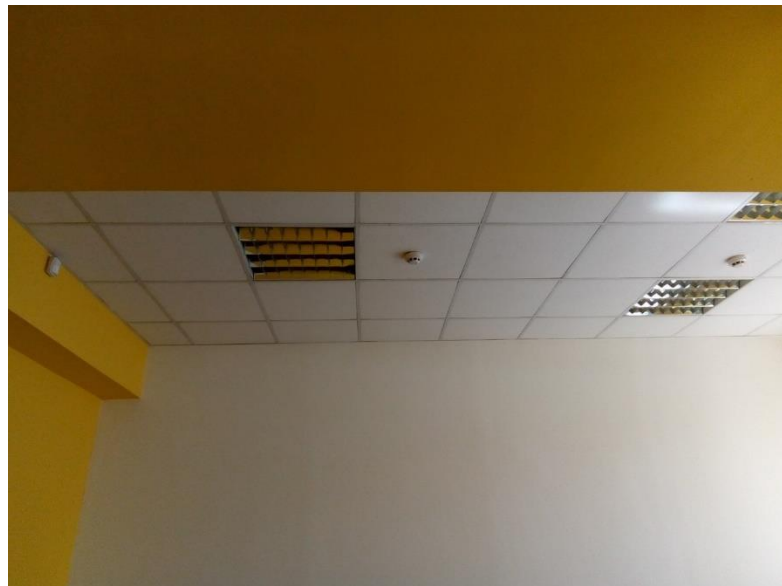


Figure 3.4 - Artificial lighting in the room

Electrical safety - is a system of organizational and technical measures and means to protect people from harmful and dangerous effects of electric current, electric arc, electromagnetic field and static electricity.

Electric hazards are able to result in burns, death and bumps :

- assume that most air wires are actually in working condition and at just high voltage. Never think that the wire is good to touch, even in case it's switched off or perhaps isolated;
- do not touch a dropping power line; a business which works with electrical energy to report the autumn of electric lines.
- stay a minimum of ten feet (three m) away from atmosphere cables during other tasks and cleaning. In case you're operating at altitudes or perhaps processing lengthy objects, examine the area before beginning job on the existence of wires;
- in case the wire gets on the car of yours while driving, stay within the automobile and continue moving from the series. In case the motor stalls, don't leave the automobile. Warn folks not to touch the automobile or perhaps wire. Call and / or ask a person to call the local electricity company and emergency services;
- never operate electric gear while standing in water;
- never restore tools or cords electric until they're qualified and not authorized;
- have a professional electrician check the power equipment which is now damp prior to activating it;
- when doing work in damp places, check out the electric cords as well as tools to ensure they're in condition that is good as well as free of defects, as well as make use of a soil fault circuit interrupter (GFCI);
- always be careful when working near electricity.

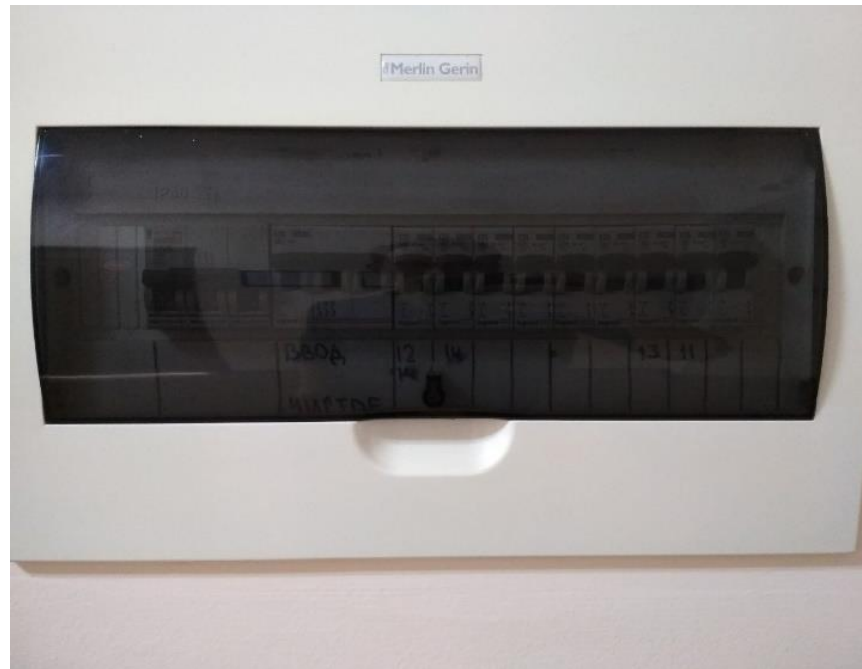


Figure 3.5 - Power distribution panel



Figure 3.6 Correct outlet of sockets

The safety of other people and staff members have to be guaranteed by the following activities:

- observance on the corresponding ranges to current carrying components or perhaps by closing, fencing of current carrying parts;
- the usage of warning signs, posters; and inscriptions;
- software of products to bring down the intensity of magnetic and electric fields to appropriate values;
- the usage of appropriate gear as well as products to guard against the consequences of magnetic and electric fields in electric installations, the place that the stress exceeds the permissible standards.



Figure 3.7 - Sources of danger in case of improper maintenance of electrical safety standard

The main insulation is the most important element of electrical installations, which determines the reliability of work and the safety of people. Isolation of live parts has the main function - to prevent the passage of electrical current by undesired paths. At the same time, it often provides protection against accidental (direct) contact with live parts. This applies primarily to wires and cables laid in residential, public and industrial buildings, as well as various types of devices used in lighting networks and electrical appliances (plug sockets, switches, fuses, lamp holders, etc.).

3.4 Fire safety

Fire safety is actually the exploration and instruction of mitigating the unwanted consequences of fires. This contains the evaluation of behavior, compartmentalization, suppression and investigation of connected fire and crisis situations, and the investigation and growth, generation, tests and software of mitigation strategies. Structures should be turned in accordance with the edition of the building code, that's legitimate when a software program for producing permits is currently being produced [14]. Building inspectors look at the compliance of the framework under construction with the building guideline. Right after the structure is really completed, the framework must be taken care of in accordance with the existing fire safety guidelines, that's furnished by the fire department of community fire department. In the event of a fire, fire investigators, other firefighters, firefighters, and emergencies are in fact known as on to mitigate, check out as well as discover out of fire harm. Lessons learned from fires are in fact put on to the improvement of all constructing needs, in addition to additions to current guidelines.

Fire safety has three effective objectives:

- continuity of operations - on a public scale, this is designed to prevent the interruption of critical services needed for public welfare;
- protection of property - on a public scale this is intended to prevent major fires in the areas. at the level of an individual building, this is usually insurance (for example, the requirement for funding) or regulatory requirements;
- safety of life - the minimum standard used in fire and building codes ;

in the course of the work, all the above-mentioned actions were considered, which would help to eliminate the emergency. all recommendations of snip rk 2.02-05-2009 "fire safety of buildings and structures" and GOST 12.1.004-91ssbt "fire safety. general requirements "

Fire safety of industrial enterprises is an important set of measures ensuring the preservation of the health of industrial workers. Such rules are developed and approved by special commissions, whose activities are aimed at preventing accidents at workplaces.

The main sources of fire in this room are electrical wiring, a computer, a projector and other electronic devices. In the case of a shortage of the wiring, it may ignite it, and as a result, the burning of neighboring flammable objects (wooden tables, fabric chairs).

One of the fire-dangerous devices is a projector. Due to the specifics of its operation, the projector quickly heats up to high temperatures and in case of failure of the cooling element it can lead to overheating and subsequent fire.

To avoid such a situation, the company has established a fire hydrant, a fire extinguisher, smoke detectors.



Figure 3.8 - Fire hydrant, fire extinguisher



Figure 3.9 - Smoke detectors

Also, in case of emergency danger, an emergency exit circuit and safety instructions are indicated in the corridor.

According to the RK standard, the fire hazard of a building can be classified as B4. This means that there are fire-hazardous items on the territory, but harm from them occurs only under certain conditions.

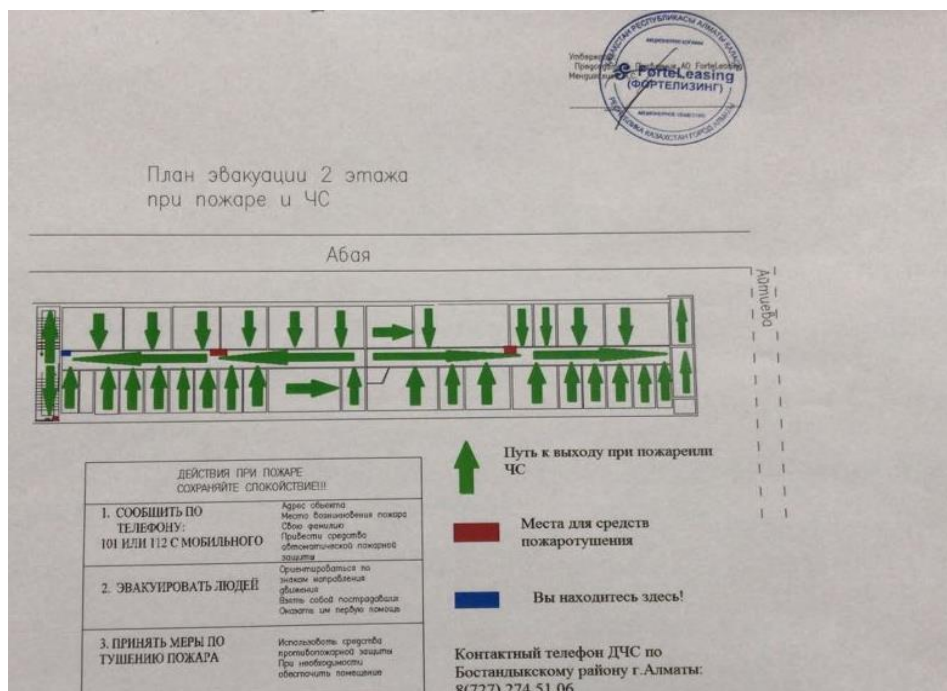


Figure 3.10 - Emergency evacuation scheme

If evacuation is not available, in case of blocking the exit, return to the room with windows (preferably a floor below). Densely bury doors, windows, ventilation. Use improvised means: furniture, cloth (clothes), to seal the cracks, and to avoid getting smoke.

An employee who uses a computer in his work may be affected by the following negative factors:

- electromagnetic and infrared radiation;
- noise of a working computer (or several computers);
- risk of electric shock in the event of a fault;
- the possibility of an outbreak.

So consider the security rules for each stage of working with the computer.

1) Before starting work: check the serviceability of the electrical wiring, the sockets and plugs of the computer, the grounding of the PC.

2) During work:

- it is necessary to handle the wires carefully;
- it is forbidden to work with a defective computer;
- you can not clean the computer when it is energized;
- it is unacceptable to carry out repair of equipment independently in the absence of special skills;
- do not place liquids near the computer, or work with wet hands;

3) In emergency situations:

- for any problems, disconnect the PC from the network immediately;

- in the event that a bare wire is detected, immediately notify all workers and exclude contact with the wire;
- in the event of a fire, take measures to extinguish it using fire extinguishers (workers should know where they are);
- In case of human injury, provide first aid and prompt medical attention.

4) Upon completion of work:

- turn off computer;
- it is desirable to conduct a wet cleaning of the workplace;
- disconnect the power supply.

Breaks during work at the computer

In order to avoid fatigue, it is recommended to take breaks lasting 10 to 15 minutes after 45 to 60 minutes of operation. During the break, the employee should perform gymnastics for the eyes and physical exercises.

Observance of the rules of work at the computer will reduce the negative impact of the computer on the health of the employee. However, most often it is the workers who neglect these rules, and the employer's task in this case is to constantly inform their employees about the consequences of non-compliance with the above requirements and order their employees to take mandatory breaks.

3.5 Calculation of ventilation

Often we are face with the question that the supply and exhaust ventilation in the office is expensive during operation due to the fact that the supply air in the winter period of the year needs to be warmed to room temperature (20-24 ° C in accordance with SNiP 2.04.05-91 *), and for most of Kazakhstan this winter period can be 8-9 months a year.

According to SNiP 2.04.05-91 * ventilation of the office premises must meet the following requirements:

- optimal parameters of air in the room
- temperature from 20 to 24 ° c;
- humidity from 35 to 60%;
- mobility of air from 0,2 to 0,3 m / s.

Calculation of the necessary air exchange must be made based on the type of room and the number of people working there. For one person the air exchange rate is from 20-60 m³ / h. This average value varies considerably depending on the purpose of the room:

- in the corridor where the long-term presence of people is not expected, this figure is reduced to 11 m³ / h;
- in the smoking room rises to 100 m³ / h;
- for a standard cabinet this value is 60 m³ / h;

To estimate the energy costs for heating the air throughout the year, you need to know the average air temperature by month (for a two-tariff meter, you need separate day and night temperatures). According to these data, the cost of energy consumption can be calculated:

The required for the exchange volume of the air is determined by the formula:

$$V_{np} = Q_{vent} * Q_{н3} / ((t_{yB} - t_{нB}) * \rho * C) \quad (3.1)$$

The temperature of the air, which is removed from the premises determined by the formula:

$$t_{yB} = t_{p3} + \Delta t * (h - n) \quad (3.2)$$

$$t_{yB} = 25 + 0,5 * (2,5 - 2) = 25,25$$

Excess heat in the premise determined by the formula:

The coefficient of energy losses for the heat sink of lighting ($E = 0.55$)

Power of one luminaire ($p = 60 \text{ W}$ (8 luminaires))

Number of windows ($m = 6$)

The area of the each window ($S = 1,5 \text{ m}^2$)

The coefficient of glazing (for double glazing) ($k = 0.6$)

Heat dissipation per person ($q = 80 \text{ W/person.}$)

Number of person ($n=5$)

$$Q_{ex} = Q_{ex1} + Q_{ex2} + Q_{ex3} \quad (3.3)$$

$$Q_{ex2} = m * S * k * Q_c \quad (6) \quad (3.4)$$

$$Q_{ex3} = x * q \quad (3.7)$$

$$Q_{ex} = (0,55 * 60 * 8) + (6 * 1,5 * 0,6 * 127) + (5 * 80) = 1349,8$$

Hence:

The volume of supply air ($Q_{vent} = 2000 \text{ m}^3/\text{h}$)

$$V_{vent} = (2000 * 1349,8) / ((25,25 - 25) * 1,197 * 1000) = 9021,2 \text{ m}^3$$

From that result, we can get ventilation rate:

The supply air temperature ($t_{vent} = 25 \text{ }^\circ\text{C}$)

The density of supply air ($\rho = 1,197 \text{ kg/m}^3$)

Specific heat of dry air ($C = 1000 \text{ J/kg}^\circ\text{K}$)

Full volume of the premise ($V_{п} = 175 \text{ m}^3$)

$$K = V_{vent} / V_{п} = 9021,2 / 175 = 51,54 \text{ times/day.}$$

3.6 Conclusion of the section

Wasteless technology is the most active form of protecting the environment from the harmful effects of emissions from industrial enterprises. The term "non-waste technology" should be understood as a set of measures in the technological processes from processing raw materials to the use of finished products, which minimizes the amount of harmful emissions to a minimum and reduces the impact of waste on the environment to an acceptable level. This set of measures includes: 1) the creation and implementation of new processes for obtaining products with the formation of the least amount of waste;

2) development of various types of drainless process systems and water-cycle cycles on the basis of waste-water treatment sewage treatment;

3) development of systems for processing waste products into secondary material resources;

4) creation of territorial industrial complexes with a closed structure of material flows of raw materials and wastes within the complex.

Today computers and office equipment are available at all commercial enterprises, banks, government agencies and public organizations. Every year, the amount of electronic equipment produced is growing exponentially. Novelties of electronics quickly supplant morally obsolete copies from desktops on distant regiments of pantries. If you cannot continue to use the computer becomes garbage. However, with electronic garbage, things are not that simple. Passive methods of environmental protection include a set of measures to limit emissions of industrial production with subsequent disposal or disposal of waste. These include: the treatment of sewage from impurities; purification of gas emissions from harmful impurities; dispersion of harmful emissions in the atmosphere; noise suppression on the way of its spread; measures to reduce the levels of infrasound, ultrasound and vibration on the ways of their spread; screening of sources of energy pollution of the environment; dumping of toxic and radioactive waste [8].

The point is that an unnecessary computer has a certain value, because contains precious and non-ferrous metals. In addition, computers have harmful substances, for example: lead and arsenic. It is for these reasons that the simple procedure of removing a computer to a trash can is illegal, and as a result, an organization may be penalized for improper operations with precious metals and environmental pollution.

4 Economical part

4.1 Technical and economic justification for the designing of the platform for distributed ledger application

The theme of the degree project focuses on the creation of an distributed ledger application, based on Hyperledger Fabric, the main aim of which is implementation of distributed ledger and blockchain technologies.

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 (4.1)

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

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
Huawei S2700-26TP-PWR-EI Mainframe	1	135867	135867
Connector RJ-11 Ship S901C	50	24	1200
Cable UTP 5e Cat 305m, 4-pair, D135-P, SHIP	1	45700	45700
Crimping pliers Ship G207	1	4207	4207
Total			186 974

The cost of equipment is provided by Ruba Technology, taking into account the exchange rate as of March 30, 2018 - 324 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]

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

Name of equipment	Quantity	Price, tenge	Cost, tenge
Laptop <u>Dell Inspiron 15.6 Series-3567</u>	2	179 990	359 980
Office table	2	15000	30 000
Chair	4	3 800	15 200
Total			405 180

Total costs for the organization of workplace: 405 180 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;

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

$$C_{el} = 0.3 \cdot 172 \cdot 19.42 = 1003 \text{ tenge.}$$

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

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

Total energy costs

$$C_{el.total} = C_{el} + C_{el.n} = 1003 + 50 = 1053 \text{ tenge.}$$

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

$$E_n = 179\,990 \cdot 0.05 = 9000 \text{ tenge.}$$

Computer depreciation will be 40% of the price

$$D_{PC} = 179\,990 \cdot 0.4 = 71\,996 \text{ tenge.}$$

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

$$C_d = 215\,500 + 71\,996 + 33\,000 + 1053 + 10\,000 = 331\,549 \text{ tenge.}$$

Table 4.3 – Calculation of costs for the development

Name of expenses	Cost, tenge
Monthly salary for the developer	215 500
Computer depreciation at development time	71 996

Monthly rent of a premise in development	33 000
Energy costs during the development	1053
Costs for additional materials and missing equipment at the time of development	10 000
Total	331 549

Calculate the capital costs by the formula (4.1)

$$C = 186\,974 + 405\,180 + 34\,680 + 331\,549 = 958\,383 \text{ tenge.}$$

Table 4.3 – Capital costs for the organization of the workplace

Name of expenses	Cost, tenge	Specific weight, %
Cost of equipment	186 974	19
Cost of workplaces	405 180	42
Installation of equipment	34 680	4
Development costs	331 549	35
Total	958 383	100

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

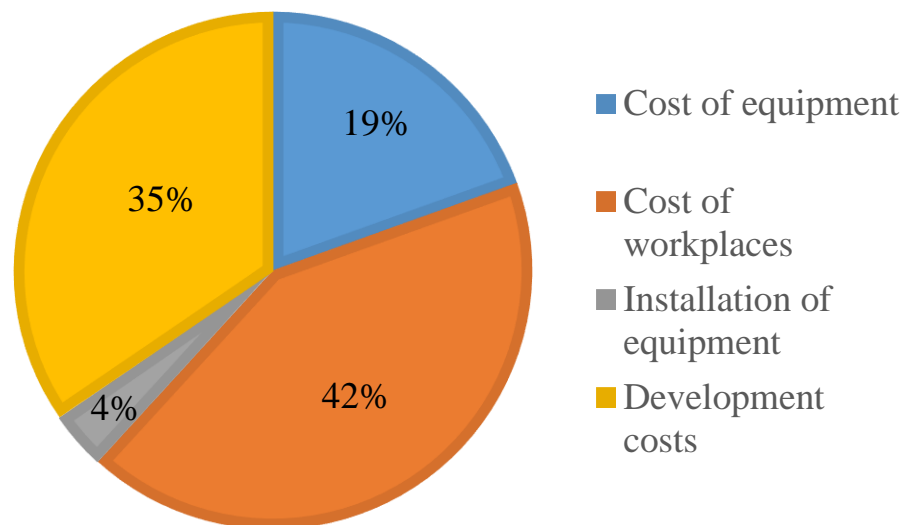


Figure 4.1 – Structure of capital costs

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 4.4 - Average salary of staff per month

List of staff	Quantity	Monthly salary, tenge	Salary per year, tenge
Engineer	2	180 000	4 320 000
Total			4 320 000

Average salary of the working staff per month is given in Table 4.4 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.

Laboratory work courses are 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:

$$W_a = 4\,320\,000 \cdot 0.3 = 1\,296\,000 \text{ tenge.}$$

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 (4.6)

$$PF = 4\,320\,000 + 1\,296\,000 = 5\,616\,000 \text{ tenge.}$$

Deductions for social tax are from 9.5%:

$$T_s = 0.095 \cdot (PF - 0.1 \cdot PF) = 480\,168 \text{ tenge.}$$

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:

$$D_1 = 359\,980 \cdot 0.4 = 143\,992 \text{ tenge.}$$

Depreciation of office furniture is 15% of the price:

$$D_2 = 45\,200 \cdot 0.15 = 6\,718 \text{ tenge.}$$

$$D = D_1 + D_2 = 143\,992 + 6\,718 = 150\,772 \text{ tenge.}$$

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 $W = 1.3$ kW;

T – hours of work for 1 year, $T = 1968$ hours per year;

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

Define the cost of electricity:

$$C_{el} = 1.3 \cdot 1968 \cdot 19.42 = 49\,684 \text{ tenge.}$$

The power consumed for necessary needs, for example light, or occasionally the inclusion of an air conditioner is calculated as 8% of the power used by the main equipment. The cost of electricity for the necessary needs:

$$C_{el.n} = C_{el} \cdot 0.08 = 3\,975 \text{ tenge.}$$

Total energy costs:

$$C_{el.total} = C_{el} + C_{el.n} = 49\,684 + 3\,975 = 53\,659 \text{ tenge.}$$

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

$$M = 405\,180 \cdot 0.05 = 20\,259 \text{ tenge.}$$

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

$$C_{adm} = 405\,180 \cdot 0.1 = 40\,518 \text{ tenge.}$$

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

$$C_{op} = 5\,616\,000 + 480\,168 + 150\,772 + 20\,259 + 53\,659 + 40\,518 = 6\,361\,376 \text{ tenge.}$$

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 4.5 – Operating costs

Operating cost Items	Cost, tenge	Specific weight, %
Payroll fund	5 616 000	88
Social tax	480 168	7.5
Depreciation deductions	150 772	2.4
Costs for materials and spare parts	20 259	0.3
Electricity costs	53 659	0.8
Other expenses	40 518	1
Total	6 361 376	100

Figure 4.2 represents the structure of operating costs.

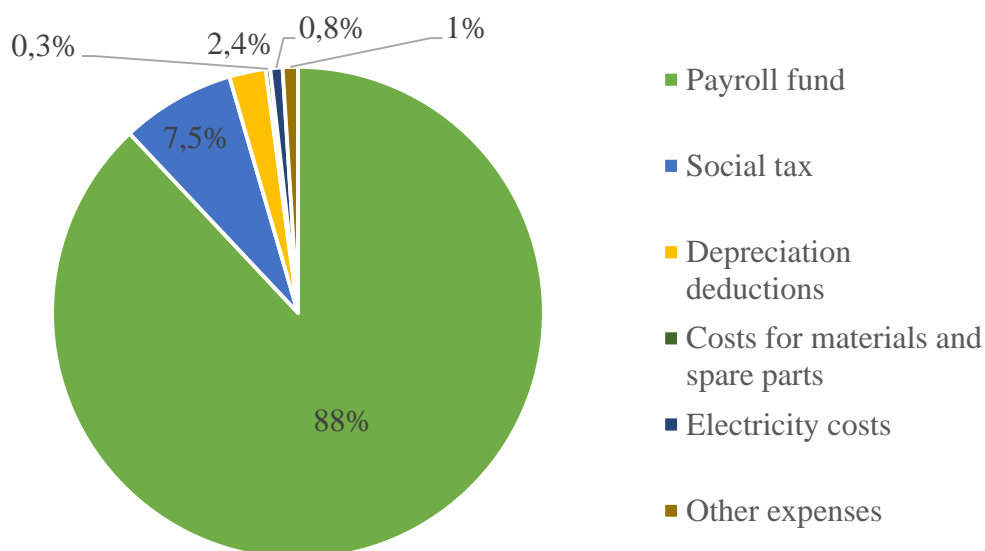


Figure 4.2 – Structure diagram of operating costs

4.4 Calculation of income

It is calculated that the courses will be 50 people, given that in the academic year is 9 months, the cost of tuition is expected to be set at 33 000 tenge per person.

$$I = 33\,000 \cdot 50 \cdot 9 = 14\,850\,000 \text{ tenge.}$$

As a result, we get that the income will be 14.85 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)

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

$$P = 14\,850\,000 - 6\,361\,376 = 8\,488\,624 \text{ tenge.}$$

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

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

$$CT = 8\,488\,624 \cdot 0.2 = 1\,697\,725 \text{ tenge.}$$

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

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

$$NP = 8\,488\,624 - 1\,697\,725 = 6\,790\,899 \text{ tenge.}$$

Absolute economic efficiency will be calculated using formula (4.12)

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

$$E = 6\,790\,899 / 958\,383 = 7.08.$$

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

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

$$T = 1/7.08 = 0.141 \text{ year.}$$

Based on the received data, we get that the costs for this project will pay off in approximately 2 months.

All the economic indicators for the project are shown in Table 4.6

Table 4.6 – Indicators of economic efficiency of the project

Indicator	Value
-----------	-------

Capital costs, tenge	958 383
Operating costs, tenge	6 361 376
Profit before taxation, tenge	8 488 624
Profit after taxation, tenge	6 790 899
Economic efficiency	7.08
Payback period, months	2 months

The usefulness of resource allocation is actually manifested from the mechanism of costs that are competitive. It takes place by doing this. Resources are sent out in such a manner that they're there to help you those firms which could greatly distribute as well as make use of them. Consumers, while agreeing to purchase solutions of firms at a greater price tag, therefore allow firms to purchase materials for generation at a greater price, in so doing diverting them from yet another, much less effective option use. As a result, materials are actually allotted in accordance with the framework of societal requirements.

Effectiveness of the usage of resources, or perhaps otherwise, creation effectiveness, is based on the motivation of firms to make optimum use of the attracted information. Those companies which use materials more badly can not retrieve the expenses of theirs at the market cost that's been started in this particular sector, and also as an effect is going to be pushed out of this particular industry. On the other hand, firms which better utilize assets will get extra cash flow (net profit) and can have the ability to boost the volume of result, as well as, appropriately, earnings.

Thinking about the economy of dynamics, it's essential to focus on the switch in economic variables. With this situation, a rationally running economic entity should continuously check the extra advantages with the extra expenses related to this particular activity. The distinction between these quantities causes it to be possible to figure out the further action. In case the marginal advantage exceeds the marginal price, the economic entity is going to continue these kinds of actions until they're identical to one another. In such a circumstance, the outcome of such steps will be optimum, as well as the aim of a rationally operating topic is going to be attained.

Taking into account all the data obtained, in the development platform for distributed ledger with capital expenditures in the amount of 958 383 tenge, net annual income will amount to 6 790 899 tenge. This project pays off for 2 months. Consequently, it can be concluded that the designing and implementation of distributed ledger – based application is cost-effective.

Conclusion

This diploma project is the first of its kind dedicated to distributed ledger technology, or blockchain. blockchain finds more and more wide application in the most various branches of a science and the industry, and this diploma project is a starting point in the further studying of the given technology. In the course of the work, the basic principles of technology were studied, such as cryptography, consensus algorithms, and popular block platforms for creating distributed applications.

As a demonstration of the possibilities of blocking technology, it was decided to create an application that would allow the entry and storage of data on the academic performance of university students. In the practical implementation section, the Hyperledger Fabric platform was chosen and its advantages were explored. The implemented application fully reflects the main principles of blockchain technology, such as decentralization and complete unchanged data entry. The functionality of the application allows you to enter data about the student, the subject and the resulting evaluation. All the data entered after confirmation by the rest of the network participants is recorded in a distributed database, after which they can not be changed. In addition, the main parameters of the constructed network were calculated, such as the time of one transaction, the number of transactions in one block, and the fault tolerance of such a network.

In the life safety activity section, factors that adversely affect employee performance were investigated. Hazardous elements that accompany work in the office were considered - fire safety and electrical threat. Optimal working conditions were also calculated, such as the calculation of artificial lighting, ventilation of the office space.

In the economic part of the project, the costs necessary for the implementation of this project were calculated, its economic efficiency was calculated, as well as the costs associated with the payment of future employees serving the network.

List of abbreviations

1. HLF – Hyperledger Fabric
2. MTBF – Mean Time Between Failure
3. HS – Hyperledger Sawtooth
4. HB – Hyperledger Burrow
5. PoW – Proof of Work
6. PoET – Proof of Elapsed Time
7. SBTF – Symplified Byzantin Fault Tolerant
8. YAC – Yet Another Consensus
9. P2P – Peer-to-Peer
10. PoS – Proof of Stake
11. RSA - Rivest, Shamir и Adleman
12. MTTR – Mean Time to Retair
13. SDK –Software Development Kit
14. API –Application Programming Interface
15. JS – Javascript
16. HTML –HyperText Markup Language
17. IoT – Internet of Things
18. JSON – JavaScript Object Notation
19. RAM - Random Access Memory
20. GPIO - General - Purpose Input/Output
21. PWM -Pulse Width Modulation
22. UART - Universal Asynchronous Receiver/Transmitter
23. SPI - Stateful Packet Inspection
24. SAW- Surface Acoustic Wave
25. IP – Internet Protocol
26. CPU – Central Processing Unit
27. TCP - Transmission Control Protocol
28. UI – User Inteface
29. CLI – Command Line Interface
30. URL – Uniform Resource Locator
31. HTTP – HyperText Transfer Protocol
32. DLT – Distributed Ledger Technologies
33. DB – DataBase
34. KYC – Know Your Customer
35. DAO - Decentralized Autonomous Organization
36. ABCI – Application Blockchain Interface
37. EVM – Ethereum Virtual Machine
38. DAH – Digital Assets Holdings

List of references

1. Seth Gilbert and Nancy Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services", ACM SIGACT News, Volume 33 Issue 2 (2002), pg. 51–59.
2. Ali M. Trust-to-trust design of a new Internet. – 2017. – p.58
3. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. – 2008. – p.47
4. Bahga A., Madiseti V. Blockchain Applications: A Hands-On Approach. – 2017.-pg. 14-18
5. Iansiti M., Lakhani K. R. The Truth About Blockchain //Harvard Business Review. – 2017. – T. 95. – №. 1. pg. 118-127.
6. 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. pg. 75-78.
7. Benchoufi M., Ravaud P. Blockchain technology for improving clinical research quality //Trials. – 2017. – T. 18. – №. 1. pg. 335.
8. Hackius N., Petersen M. Blockchain in logistics and supply chain: trick or treat?. – epubli, 2017.
9. Cretarola A, Figà-Talamanca G, Patacca M. A sentiment-based model for the BitCoin: theory, estimation and option pricing. (2014) p 6.
10. Garcia D, Tessone CJ, Mavrodiev P, Perony N. The crypto traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy. Journal of the Royal Society Interface. (2013) p.7.
11. Geweke, J. Measurement of linear dependence and feedback between multiple time series. Journal of the American statistical association,(1982). p.304-313.
12. Hafner, C. Cryptocurrencies with time-varying volatility(2018) p.5
13. Hayes AS. Cryptocurrency Value Formation: An empirical study leading to a cost of production model for valuing Bitcoin. Telematics and Informatics. (2016). p11.
14. Kristoufek, L. What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. (2015, October) p.12.
15. Kroll, J. A., Davey, I. C., & Felten, E. W. (2013, June). The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In Proceedings of WEIS (Vol. 2013) p.13.
16. Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of ddos attacks against bitcoin mining pools. In International Conference on Financial Cryptography and Data Security pages 72–86. Springer, 2014.
17. Virtual Box Cloud Software. // virtualbox.org //Сервер компании Virtual Box. 2017. URL: <https://www.virtualbox.org/> (date of the request: 15.02.2017).

18. K. Hooda S., Kapadia Sh., Krishnan P. Using TRILL, FabricPath, and VXLAN: Designing Massively Scalable Data Centers (MSDC) with Overlays. – N.: Cisco Press, 2014. – 127с
19. W. Del Signore K. Measuring and Simulating Cellular Switching System IP Traffic // Bell Labs Tech Journal. – 2014. -№4. – С 159-180
20. Баклашов Н.И., Китаева Н.Ж., Терехов Б.Д. Охрана труда на предприятиях связи и охрана окружающей среды: Учебник. – М.: Радио и связь, 2008. – 385 с
21. Т.М.Попова, Т.В.Ходанова. Дипломное проектирование. Методические указания к выполнению экономической части.- Алматы: АИЭС. 2000.-27 с.
22. Hyperledger-fabricdocs Docomentation //Linux Foudation – 2015.p. 58-141
23. <https://blockchain-fabric.blogspot.com/2017/04/hyperledger-fabric-v10-block-structure.html> (date of request: 5.05.2018)
24. <https://stackoverflow.com/questions/50225696/what-is-the-maximum-transaction-size-in-hyperledger-fabric>(date of request: 8.05.2018)
25. <https://stackoverflow.com/questions/40954717/what-is-the-size-of-a-block-in-hyperledger-v0-6>(date of request: 12.05.2018)
26. Blockchain. Research paper. A. Shanti Bruyn, 2017. p. 78
27. Е.Хакимжанов. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002 - 30 стр
28. Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности: Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат - Алматы: АИЭС, 2009. - 20 с
29. Базылов К.Б., Алибаева С.А., Бабич А.А. : Методические указания по выполнению экономического раздела выпускной работы бакалавров для студентов всех форм обучения специальности 050719 – Радиотехника, электроника и телекоммуникации – Алматы: АИЭС, - 2008. -19 с.
30. Кодекс Республики Казахстан от 10 декабря 2016 года № 99-IV «О налогах и других обязательных платежах в бюджет (Налоговый кодекс)» (с изменениями и дополнениями по состоянию на 28.04.2016 г.), ст.120.

Appendix A

Listing of the startFabric.sh program

```
#!/bin/bash
#
set -e

# don't rewrite paths for Windows Git Bash users
export MSYS_NO_PATHCONV=1

starttime=$(date +%s)

if [ ! -d ~/.hfc-key-store/ ]; then
    mkdir ~/.hfc-key-store/

# launch network; create channel and join peer to channel
cd ../basic-network
./start.sh

# Now launch the CLI container in order to install, instantiate chaincode
# and prime the ledger with our 10 tuna catches
docker-compose -f ./docker-compose.yml up -d cli

docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/
hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp" cli
peer chaincode install -n tuna-app -v 1.0 -p github.com/tuna-app
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/
hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp" cli
peer chaincode instantiate -o orderer.example.com:7050 -C mychannel -n tuna-app -v 1.0 -c '{"Args":[""]}'
-P "OR ('Org1MSP.member','Org2MSP.member')"
sleep 10
docker exec -e "CORE_PEER_LOCALMSPID=Org1MSP" -e "CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/
hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp" cli
peer chaincode invoke -o orderer.example.com:7050 -C mychannel -n tuna-app -c
'{"function":"initLedger","Args":[""]}'

printf "\nTotal execution time : $((($(date +%s) - starttime)) secs ...\n\n"
printf "\nStart with the registerAdmin.js, then registerUser.js, then server.js\n\n"
```

Appendix B

Listing of the registerUser.js program

```
'use strict';

var Fabric_Client = require('fabric-client');
var Fabric_CA_Client = require('fabric-ca-client');

var path = require('path');
var util = require('util');
var os = require('os');

//
var fabric_client = new Fabric_Client();
var fabric_ca_client = null;
var admin_user = null;
var member_user = null;
var store_path = path.join(os.homedir(), '.hfc-key-store');
console.log(' Store path:'+store_path);

// create the key value store as defined in the fabric-client/config/default.json 'key-value-store'
setting
Fabric_Client.newDefaultKeyValueStore({ path: store_path
}).then((state_store) => {
    // assign the store to the fabric client
    fabric_client.setStateStore(state_store);
    var crypto_suite = Fabric_Client.newCryptoSuite();
    // use the same location for the state store (where the users' certificate are kept)
    // and the crypto store (where the users' keys are kept)
    var crypto_store = Fabric_Client.newCryptoKeyStore({path: store_path});
    crypto_suite.setCryptoKeyStore(crypto_store);
    fabric_client.setCryptoSuite(crypto_suite);
    var tlsOptions = {
        trustedRoots: [],
        verify: false
    };
    // be sure to change the http to https when the CA is running TLS enabled
    fabric_ca_client = new Fabric_CA_Client('http://localhost:7054', null, '', crypto_suite);

    // first check to see if the admin is already enrolled
    return fabric_client.getUserContext('admin', true);
}).then((user_from_store) => {
    if (user_from_store && user_from_store.isEnrolled()) {
        console.log('Successfully loaded admin from persistence');
        admin_user = user_from_store;
    } else {
        throw new Error('Failed to get admin.... run registerAdmin.js');
    }
    return fabric_ca_client.register({enrollmentID: 'user1', affiliation: 'org1.department1'},
    admin_user);
}).then((secret) => {
    // next we need to enroll the user with CA server
    console.log('Successfully registered user1 - secret:'+ secret);
    return fabric_ca_client.enroll({enrollmentID: 'user1', enrollmentSecret: secret});
}).then((enrollment) => {
    console.log('Successfully enrolled member user "user1" ');
    return fabric_client.createUser(
        {username: 'user1',
        mspid: 'Org1MSP',
        cryptoContent: { privateKeyPEM: enrollment.key.toBytes(), signedCertPEM: enrollment.certificate }
    });
}).then((user) => {
    member_user = user;
    return fabric_client.setUserContext(member_user);
}).then(()=>{
    console.log('User1 was successfully registered and enrolled and is ready to intreact with the fabric
network');
}).catch((err) => {
    console.error('Failed to register: ' + err);
    if(err.toString().indexOf('Authorization') > -1) {
        console.error('Authorization failures may be caused by having admin credentials from a
previous CA instance.\n' +
        'Try again after deleting the contents of the store directory '+store_path);
    }
});
```

Appendix C

Listing of the server.js program

```
//SPDX-License-Identifier: Apache-2.0

// nodejs server setup
// call the packages we need
var express      = require('express');           // call express
var app          = express();                   // define our app using express
var bodyParser   = require('body-parser');
var http         = require('http')
var fs           = require('fs');
var Fabric_Client = require('fabric-client');
var path         = require('path');
var util         = require('util');
var os           = require('os');

// Load all of our middleware
// configure app to use bodyParser()
// this will let us get the data from a POST
// app.use(express.static(__dirname + '/client'));
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

// instantiate the app
var app = express();

// this line requires and runs the code from our routes.js file and passes it app
require('./routes.js')(app);

// set up a static file server that points to the "client" directory
app.use(express.static(path.join(__dirname, './client')));

// Save our port
var port = process.env.PORT || 8000;

// Start the server and listen on port
app.listen(port, function(){
  console.log("Live on port: " + port);
});
```