Composition of Software through Components using Blockchain Technology

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

in

Computer Science and Engineering

by

Shivangi Baranwal

2018CS09

Under the Supervision of

Prof. D.K. Yadav

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY ALLAHABAD,

PRAYAGRAJ – 211004, INDIA

June 22, 2020

I declare that the work presented in this thesis titled "***Composition of Software through Components using Blockchain Technology*** ", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj for the award of the ***Master of Technology*** degree in ***Computer Science and Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

June 22, 2020
Prayagraj

(Shivangi Baranwal)

CERTIFICATE

Certified that the work contained in the thesis titled "***Composition of Software through Components using Blockchain Technology***", by *Shivangi Baranwal*, Registration Number *2018CS09* has been carried out under my supervision and this work has not been submitted elsewhere for a degree.

(Prof. D.K.Yadav)
Computer Science and Engineering Dept.
MNNIT Allahabad, Prayagraj

June 22, 2020

Acknowledgment

I would like to take this opportunity to express my deep sense of gratitude to all who helped me directly or indirectly for my thesis work. Firstly, I would like to thank my supervisor, **Prof. D.K. Yadav**, for his advice and encouragement behind the successful completion of this dissertation. The confidence shown in me by him was the most significant source of inspiration for me. He provided me ample opportunities to explore myself. Besides my advisor, I would like to thank the rest of my thesis committee member: Dr. Ranvijay and Dr. V.S. Tripathi, for their insightful comments and encouragement, but also for the hard question which incite me to widen my research from various perspectives.
I wish to express my sincere gratitude to **Prof. Rajeev Tripathi,** Director, MNNIT Allahabad, Prayagraj, and **Prof. Anil Kumar Singh,** Head, Computer Science and Engineering Department, for providing me with all the facilities required for the completion of this thesis work.
Last but not the least, I would like to thank my family: my parents and my brothers for supporting me throughout my life and also help me morally to write this thesis.

Shivangi Baranwal

Biographical Sketch

Shivangi Baranwal

Address- Azad Nagar North, Barhaj (Deoria), U.P.- 274601
E-Mail: shivangibaranwal.mnnit@gmail.com, Contact No.: +91-8115145699

Education

- Pursuing M.Tech. in Computer Science and Engineering from MNNIT Allahabad, Prayagraj with CPI of 8.0/10.0.
- B.Tech. in Information Technology from Dr. A.I.T.H. Kanpur affiliated by Dr A.P.J. Abdul Kalam Technical University, Lucknow with 76.3% in 2017.
- Intermediate from U.P. Board with 68% in 2011.
- High School from U.P. Board with 72% in 2009.

– Dedicated to my loving family for their kind love and support and my thesis supervisor Prof. D.K. Yadav for sharing his valuable knowledge, encouragement and showing confidence in me all the time.

❝To succeed in your mission, you must have single-minded devotion to your goal.❞

-Dr. A. P. J. Abdul Kalam

Synopsis

Blockchain is an emerging technology for establishing trust and security in various applications other than cryptocurrency. Nowadays, user's requirements change frequently and because of these changes, there is need for complex softwares. To overcome this challenge, the software community need revolutionary changes to move from traditional approaches to component-based software development technique.

In component-based approach reusable components are used to develop a software system by selecting the appropriate software components as per the requirements specification and assemble all the components with well-defined software architecture. Many researchers have proposed numerous algorithms, models and techniques for developing software components and also various architectures for the composition of software by using components from software component repository. The selection of expected third party software components to develop the software is not always trustworthy. So, there is a need for certified components with proper standardization. Missing quality of assurance and standardization of components led to propose a technique that assures the validation of components.

Blockchain technology provides trust to software developers as well as to users. The idea is to propose a technique based on blockchain to overcome the standardization problem of components for giving quality assurance to the software. By using this technique, we can easily construct the software in place of developing the software.

Contents

List of Figures

Abbreviations

CBSE	Component based Software Engineering
COTS	Component Off The Self
IPFS	InterPlanetary File System
P2P	Peer-to-Peer

Chapter 1

Introduction

Nowadays our software developer gives priority to the concept of component Based Software Engineering (CBSE) to construct the software. Component is a modular, portable, replaceable, and reusable set of well-defined functionality that encapsulates its implementation. In this concept, complex software systems are assembled by using already existing software components in place of writing codes from scratch. It promotes the reusability of existing components, decreases time and cost [1]. By the procedure of CBSE, we can integrate and upgrade our system in the future just by the deployment of some components in place of disturbing the whole system.

Components are purchased from different vendors also known as Commercial Off The Shelf components (COTS) or from the software component repositories according to domain specification. After selecting required components, they are assembled to build software [5]. The developer of components and constructor of software by using these components could be different so there is a need for a technology which would give certification and standardization to make components trustworthy. For developing quality assured and standardized components, we are going to use blockchain technology.

Blockchain technology is also known as Distributed Ledger Technology (DLT). Blockchain

is a decentralized and trustworthy digital public immutable interconnected ledger. It contains several blocks. These blocks are interconnected by hash cryptography. Every Block is immutable so there is no possibility of malicious attacks in the future. This technology provides transparency, trust, reliability and security. So the software developer can trust on existing components for building the software as well as Users will also trust on the standardized composite software.

1.1 Challenges of CBSE

CBSE is facing many challenges today, some of these are summarised in the following [8]:

- Trusted components
- Component certification
- Component selection and requirement management
- Component configurations
- Long-term management of component-based system
- Composition predictability
- Tool support

Component development is done by a third party. Users of the components do not know the reliability and robustness of the component. There is no measurement of standardization and trustworthiness of the component.

A certificate is a standard processor in many fields yet it is not launched in software components. Certification will provide trust and quality to components as well as software that would be constructed by the certified components.

There is a need for domain engineering to manage requirements and selection of components for building software. Many algorithm are proposed for this purpose but which one should be selected, is a complex problem. The method should fulfill the main requirement of CBSE that is the reusability of components.

There could be a need for complex software, so to configure all components based on version and functionality is not an easy task. Communication of components is done by interfaces. There are some proposed models to fulfill this, yet lots of improvements are needed. Technical issues, administrative and organizational issues, legal issues are some long term issues in CBSE which should be managed properly.

The objective of CBSE is to build systems from reusable components simply and efficiently. This can only be achieved with extensive tool support. Some examples of tools are component selection and evaluation tools, tools for managing the repositories, component testing tools, component-based design tools, component configuration tools, run-time system analysis tools, etc.

1.2 Motivation

The traditional method to develop software is done from scratch by writing code [4], which is time taken and costly. Component-Based Software Engineering (CBSE) is done by reusability of components. The reusability of components decreases cost and time, and also increases quality and productivity. CBSE is mainly the construction of software in place of development.

In the challenges of the component-based software development process, we can see the problem of certification of components and trustworthiness of software build by using components. Quality assurance and validation of every single component will increase trust among software engineers as well as users. Quality assurance is done by black box testing and white box testing in the software testing phase, but it is not so effective.

nowadays, blockchain technology is becoming popular for security and trust purposes in many companies, like: Walmart, IBM, many financial industry, Supply chain management related industry and in real estate business etc. Blockchain is a decentralized and trustworthy digital public interconnected ledger. It uses distributed techniques and consensus algorithms that were maintained by all participants. It saves time and money, reduces risk and increases trust. So, for making trustworthy components and giving standard to CBSE, blockchain technology can be used.

1.3 Objective

There are many phases for the composition of software through components. One of the phases is software certification. In CBSE, the software is constructed by using components. So, there is necessary to validate every component. These days, complex software are developed by a software developer. It takes time, cost and lots of effort. So, certification and quality assurance of components and of software, will make worth to all the efforts of developers.

The main objective is to construct the software rather than develop the software.

- Creation of software repository
- Use of blockchain to provide trustworthy components
- Composition of software from already build components

We work in this thesis to achieve the goal of providing quality and standardization to components for making trustworthy software by the composition of the components without having any negative effect on the basic objective of CBSE.

1.4 Summary

In this chapter, we saw a brief overview of CBSE and its main problem (quality assurance of component), Blockchain technology and it's features that satisfy the demand to solve the problem of CBSE. Also, we have a little idea about what is the purpose of this thesis work and why it is required.

1.5 Organization of The Thesis

The rest of the thesis is organized as follows:

Chapter 1 Contains introduction, overview of CBSE and blockchain technology and motivation of the thesis.

Chapter 2 Contains background study and literature survey which acknowledges previous work in the proposed field.

Chapter 3 Describes proposed work with required algorithm.

Chapter 4 Explains the implementation process and results.

Chapter 5 Finally concludes the thesis work and provide future research directions.

Chapter 2

Background and Literature Survey

2.1 Introduction

In this chapter we will learn about Component Based Software Engineering (CBSE), advantages of CBSE, life cycle of component based software development, different models and concepts of CBSE.

After review of many research papers I decided to work on quality assurance and standardization of CBSE. For this, I chose Blockchain Technology because it is trustworthy and will provide certification to components. So, we will also learn about blockchain technology, crypto-currency, important features of blockchain, block structure and schemes that support blockchain technology.

2.2 Components Based Software Engineering (CBSE)

Over many years, the huge uses of software have new demands and expectations on the software industry especially of time and cost saving software for improving quality and

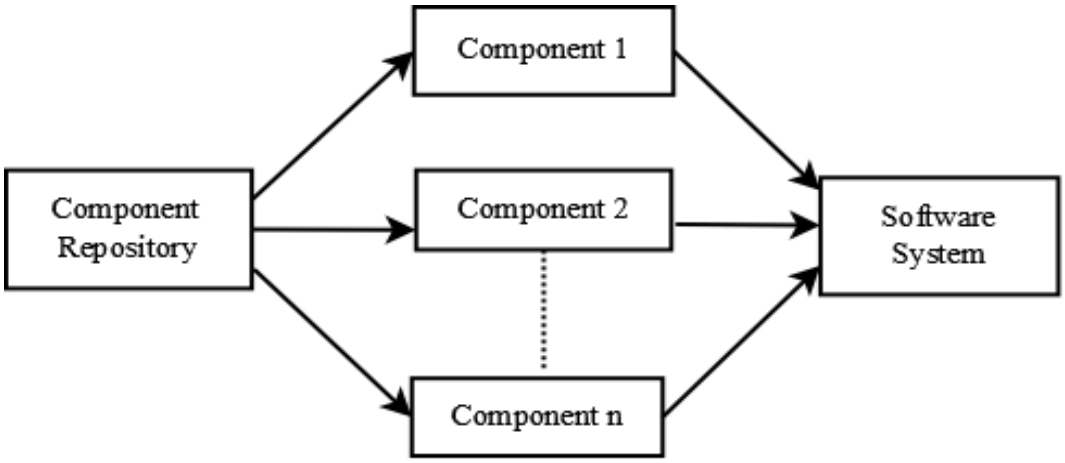

Figure 2.1: Component Based Software Development [4]

enhancing the development productivity. Such concepts can be achieved through the organized approach of reusability with Plug and Play methodology using Component Off The Shelf (COTS) [5]. According to Szyperski's et al. [4] component definition is 'A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties'. A component is an independent executable unit that has already been tested and deployed. So, every component should be verified and validate to maintain standardization. Then these generic components are selected from a repository and these selected components are assembled together to form a new software [5].

2.3 Advantages of CBSE

Advantages of CBSE are described below:

- **Component reusability**: As we know every component works as a plug and play device [4], so it can be used repeatedly. Domain engineering [2] helps to increase the reusability of a single component.

- **Decrease time and cost**: Because of the reusability scheme, we can construct the complex software in less time, cost and efforts.

- **Increase quality product**: As it takes less time and cost, productivity is increased and because of reusing of software components that are already tested and validated, because of this, quality of the product is increased. In general, there is a need to verify the component every time, blockchain solves this problem because when a block is created and validated, there is no need to verify it every time.

- **Decrease complexity**: The meaning of decreasing complexity in CBSE is reusing components with encapsulated function and providing standardized rules for component composition through component interfaces [5].

- **Increasing maintainability**: Components can be enhanced in the future as per according to need so that we can change that particular component and maintain as per user requirement. This is also useful when a new version of a software is released.

- **Increase efficiency**: In CBSE, there is no need to build software from scratch. By using components, the development becomes more efficient because of the reusability and quality assurance of components.

2.4 Component Models

components are used to build the integrated application software by using the list of component models. Component models prescribe the standards and deliver the interfaces between the component and the application software [5]. Some component models are depicted below:

- Microsoft COM, DCOM (Component Object Model, Distributed COM)

- Object Request Broker Architecture (CORBA) by OMG

- Oracle's Enterprise Java Beans (EJB)

2.5 Life Cycle of Component-Based Software Systems

Component-based software systems are developed by selecting various appropriate components and assembling them in place of programming an overall system from scratch. Thus the life cycle of component-based software systems are something different from traditional software systems. The life cycle of component-based software is divided into two parts which run in parallel. These are:

1. Component-based software development life cycle
2. Component development life cycle

Description of these two life cycle is given below [4]:

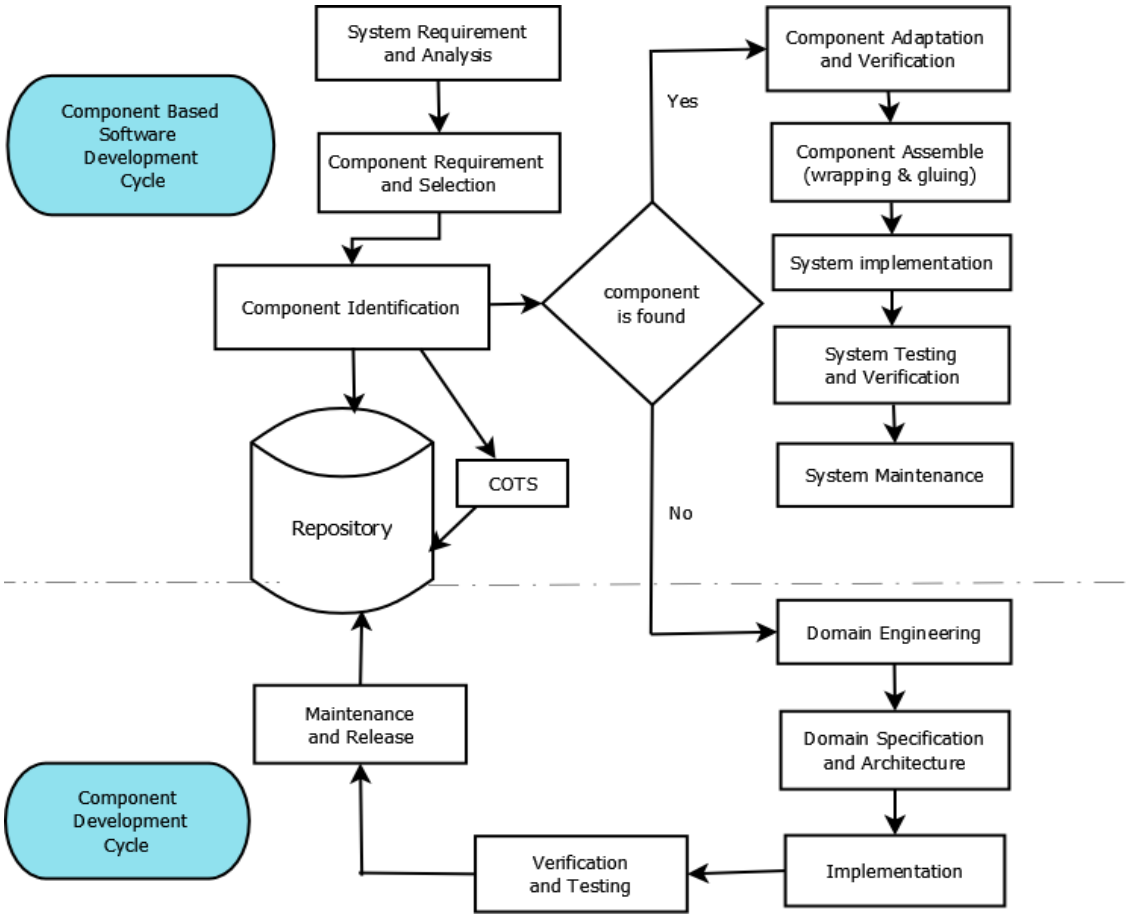

Figure 2.2: Life Cycle of CBSE [7]

1. **Component Based Software Development Life Cycle**
   When a component is found in the component repository, following steps happen:

   (a) Component Requirement Analysis

   (b) Component Development

   (c) Component Certification

   (d) Software architecture selection design

   (e) System Integration

   (f) System Testing

   (g) System Maintenance

2.5.1 Component Requirement Analysis

component requirement analysis consists of four main steps: requirements gathering and definition, requirement analysis, component modeling, and requirement validation [4]. In this phase, we also take care of programming language, the platform and the interfaces related to the component. Domain engineering or product line engineering [2] is helpful in this phase. Domain engineering means the selection of all components from a repository should be of the same domain. The output of this phase is the current requirement documentation.

2.5.2 Component Development

The input of this phase is the output of the previous phase i.e. requirement documentation. Component development is the process for fulfilling the requirements for a well-functional, high-quality component with multiple flexible interfaces. It provides well-developed components and documents for further phases.

2.5.3 Component Certification

Component certification is important for providing quality assurance to system software engineers as well as to users. It consists of the following points:

- **component outsourcing**: managing a component outsourcing contract and auditing the contractor performance
- **component selection**: selecting the right components per the requirement for both functionality and reliability
- **component testing**: confirm the component satisfies the requirement with acceptable quality and reliability

2.5.4 System Architecture Design

It consists of software architecture selection, construction, analysis, and evaluation through components. The objectives of system architecture design are to collect the users' requirements, identify the system specification, select appropriate system architecture, and determine the implementation details such as platform, programming languages, etc.

2.5.5 System Integration

System integration is the process of assembling components selected into a whole system under the designed system architecture. There are four steps in this phase: integration, testing, changing component and reintegration (if needed).

2.5.6 System Testing

The objective of system testing is the final system integrated by components selected in accordance with the system requirements. System testing should contain function testing and reliability testing.

2.5.7 System Maintenance

The objectives of system maintenance are to provide an effective product or service to the end-users while correcting faults, improving software performance or other attributes, and adapting the system to a changing environment.

2. **Component Development Life Cycle**
   When a component is not found in the repository, it is developed from scratch. Steps of component development are given below:

   (a) Domain Engineering, Specification and Architecture

   (b) Implementation

   (c) Packing

   (d) Verification and Testing

   (e) Deployment and Release

2.5.8 Domain Engineering, Specification and Architecture

The main concern of domain engineering is the development of reusable software. While developing a component for the same domain of software, domain engineering becomes helpful. The main aim of domain engineering is to have a generic system or component that can be used efficiently in different systems with little or no change.

The component specification states different properties that are important for corresponding component implementation. Some of the important properties are functionality, behavior, Interaction potential and Quality properties (performance, portability and stability).

2.5.9 Implementation

The main concern of this phase is to develop a generic component which can be easily reusable and adaptable in similar other domain. The study of the framework

is required as a component usually developed based on an architectural framework and follows models like- Microsoft COM, DCOM (Component Object Model, Distributed COM). Object Request Broker Architecture (CORBA) by OMG. Oracle's Enterprise Java Beans (EJB). etc.

2.5.10 Packing

In this phase, all the reusable components are given a rank based on the percentage of reuse and based on cost-benefit analysis. Ranking helps the concerned person to identify and select the components.

2.5.11 Verification and Testing

This is a very essential phase of component development for reuse. In this phase verification and testing of the component are being done to know the capability and accuracy of the component. Unit testing, black box testing and white box testing are done for this purpose.

2.5.12 Deployment and Release

This is the last phase for developing software. After this, that new component is stored in a repository for performing the composition of software through components.

2.6 Quality Assurance in CBSE

In the software certification phase [4], validation of every component is performed. It is done by black-box testing and white box testing [6]. But these are not sufficient approach to maintain standardization and trust. Blockchain technology provides transparency, trust, reliability and security. For quality assurance of component development, blockchain technology can be used.

2.7 Blockchain Technology

Blockchain was first invented by Stuart Haber and W.Scott Stornetta in 1991. But it is introduced by Satoshi Nakamoto in 2008 in the form of cryptocurrency, Bitcoin, through the maintenance of immutable distributed ledgers in thousands of nodes [21]. It is also known as Distributed Ledger Technology (DLT).

Blockchain is a decentralized and trustworthy digital public interconnected ledger. It uses distributed techniques and consensus algorithms that were maintained by all participants. It saves time and money, reduces risk and increases trust.

There are many cryptocurrencies used in market like:- bitcoin [21], litecoin, ethereum [20], z-cash, dash, ripple etc. The most popular cryptocurrency is bitcoin. Bitcoin provides a peer-to-peer payment network of electronic cash without any intervention of a third party. It uses SHA-256 mathematical hash function for encrypting every block. People invest in bitcoin without knowing each other because they trust each other as they are using blockchain technology. It is predefined that 21 million bitcoin can exist in the bitcoin protocol. The halving policy is used to fulfill this target by 2140. Bitcoin is becoming popular day by day because it overcomes the problems of traditional banking. Problems with banking transaction systems are the technical issue, accounts can be hacked, transfer limits and extra charges.

Blockchain applications could be divided into three stages: Blockchain 1.0, 2.0, and 3.0.

- Blockchain 1.0 is the deployment of cryptocurrencies as a peer-to-peer cash payment system.
- Blockchain 2.0 is the extensive blockchain applications than simple cash transactions, including stocks, bonds, loans and smart property.
- Blockchain 3.0 is developing blockchain applications beyond currency, finance, and markets, such as in the areas of government, health, science, literacy, culture, and art.

Block is a collection of recent and verified transactions. It is the smallest unit of a blockchain. For the addition of a new block into a blockchain, a verification process is required called consensus.

Technical background of blockchain technology are:

2.7.1 Minors

These are the nodes that participate in the competition of validating transactions and creating blocks from them to get incentives related to that transaction validation [28]. Minor's main job is to add a block to the blockchain which can be done by solving a mathematical puzzle that requires higher computational power. Minor who adds a block in the blockchain gets all the rewards that are related to the validation of transactions and blocks.

2.7.2 Consensus Algorithm

It is a phenomenon in which all or majority of blockchain nodes have agreement on a particular massage or task. In the blockchain, we use different algorithms to verify if consensus is reached. It is useful to elect a leader who will decide the content of the next block. This leader has the responsibility of broadcasting just created block to the network. Two different consensus algorithms are Proof of Work (PoW) and Proof of Stake (Pos).

2.7.3 Proof of Work (PoW)

In this consensus algorithm, a leader is elected based on solving a mathematical puzzle that requires higher computational power. Every miner gets different puzzles for each block with the same difficulty. In this problem, he has to find a hash output for data in its block which will start with certain zeros, which is called target. A hash function is simply a mathematical problem that is very hard to solve, but very easy to verify the answer. To solve the problem, if data of block doesn't hash it into the required output string which starts with few leading zero's, a miner needs to change a data field repeatedly inside a block called nonce with the range of 0-4 Billion. By changing nonce after many computation miners will able to get the output hash. Then miner broadcasts his hash to all the nodes and all nodes provide their views to reach consensus by verifying the solution [28]. All this process requires a lot of electricity as well as computational power consumption.

2.7.4 Proof of Stake (PoS)

It overcomes shortcomings of the PoW. Here miner makes a bid by keeping a few parts of their resources as stake and get elected as a leader or validator. A miner making the highest bid will get the highest share of validating transactions and rewards related to them. This algorithm is biased towards rich people, but it stabilizes the system by avoiding the possibility of 51 percent attack which can be possible in the case of PoW. In this case, it is not possible because executing an attack might result in the loss of a validator's stake and also as he has the resources in the blockchain, so it will result in loss of attacker from his attack. So, this attack is not possible as it works like self-destructive mode.

Zcash offers better privacy and selective transparency of transactions by using a proof-of-zero-knowledge consensus algorithm [22].

2.7.5 Smart Contract

These are digital contracts that can control the user's digital assets, formulate the participant's right and obligation, will automatically execute by the computer system. It's not just about a single node procedure, it is seen jointly of contract participants, can respond to message what it receive and store the data, it can also send message or price to outside. Sensible Contract is just like an individual is trusty, will hold the assets quickly and can follow the order that has already been programmed. Ethereum associates open-supply blockchain platform by combining sensible Contract, by providing suburbanized virtual machine to handle the contract, by victimization its digital currency referred to as ETH. people produce various services, applications or contracts on this platform.

2.8 Advantage of Blockchain Technology

- **cryptographic hash function (SHA-256)**: Here SHA stands for Secure Hash Algorithm. This algorithm is mostly used to prove data integrity. This is similar to the unique signature for a particular text or data file. It generates almost 256 bit (32 Bytes) unique signature for the text. The same input will always produce the same output. Multiple hash values are stored in the form of a Merkle tree. SHA-256 is

a strong cryptography algorithm because it is one way, deterministic and has fast computational speed. It also has a rare match because of the avalanche effect.

- **Decentralization**: Blockchain technology is independent of centralized node/server. There is no problem of single failure system. Each node behaves as a client as well as a server. Operations on the data can be performed in a distributed manner.

- **Immutability**: The record stored once cannot be modified later. These records are permanently stored and linked by cryptographic hash. Data tampering can be encouraged only in the case when an attacker takes control of more than 51 percent nodes of the distributed environment at a particular time. Blockchain technology succeeded in gaining the trust of the user and because of this a node can transfer or make transactions to other nodes by just knowing its public address. This feature encourages the global acceptance of this technology.

- **Transparency**: The same ledger or records are stored at every node in the distributed environment. Any changes are verified and reflected in all the nodes.

- **Open source**: Most of the blockchains are public in nature and its source code is available for use. All the records are public while user identities are encrypted/hidden. We can use blockchain technology to developing any application according to our needs.

  Blockchain can be developed in either a private way or a public way and can be served as an open or closed way [24]. Like e-voting system is public and closed, video games and cryptocurrencies are public and open, Zcash payments are published on a public blockchain, but the sender, recipient, and amount of a transaction remain private. Military, National defense, law enforcement, tax returns are private and closed. Supply chain management, some government sectors are private and open.

2.9 Structure of Blockchain

Generally, blockchain can be considered as a singly linked list in which pointer pointing to the next node/block can be linked dynamically (i.e. changes reflected to change in the hash) and a new node/block will only get added to the end of the linked list. The blockchain contains multiple blocks linked by cryptographic hash and each block contains main data, magic number, block size, version, hash of previous block, hash of the current block, Merkle root, nonce, timestamp and other information.

Block: It is a collection of records or transactions made by uses and is residing in the pool of unconfirmed transactions. All ledgers are stored in the block and all blocks are linked to one another in which previous blocks are immutable and tamper-proof. The header part of the block contains block id, timestamp, nonce.

Main Data: It is dependent on the type of service or application. For example transaction records, voting ballots, healthcare data, IoT data records, etc.

Hash of the previous block: Blocks are interconnected by hash cryptography. Every block has an entry of previous hash by which it is connected to the previous block and hash of the current block will be the previous hash of the next block. This field establishes a link with the previous block and maintains the integrity of the blockchain.

2.10 Life cycle of transaction in Blockchain

Here I have described the transaction of a blockchain as a seven-step process. By assuming the test case of bitcoin cryptocurrency, a step by step procedure followed is given below:

Step 1: User A wants to send some amount of bitcoin to User B. Then user A should know the public address of user B to send money (User B requested for money in the network by sharing his public key).

Step 2: This transaction made by user A goes to the 'Pool of unconfirmed nodes' where all the unconfirmed or newly made transaction resides.

Step 3: Here comes the role of miners (mostly referred to nodes) in the network. They pick the transactions in order of their order of increasing timestamp and add them to their block (collection of transactions) with some additional information about the block.

Every miner has the right to create their block, but multiple miners can add the same transaction into the blocks created by them. For example, if there are two miners A and B. Both of them decided to include transaction T into their block. Before adding any transaction to the block, the miner needs to verify them. They should check whether these transactions are in the order for execution according to the blockchain. This verification is done by checking the supposed balance of the sender on the blockchain. The sender should have sufficient balance to get added to the blockchain.

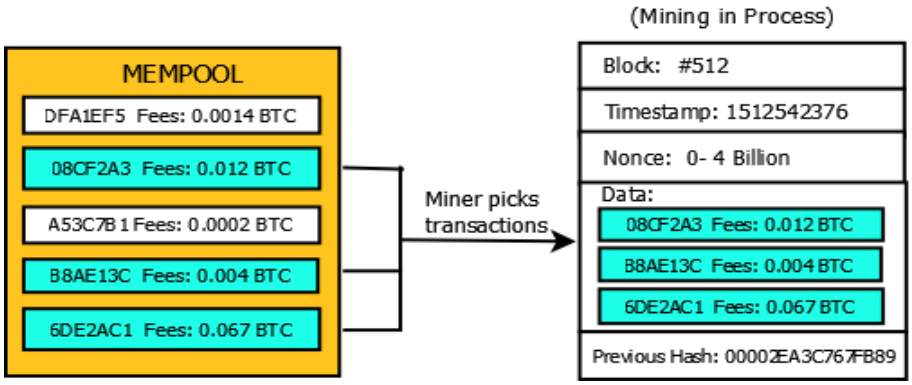

Figure 2.3: Mining Process

Step 4: After filling the block with verified transactions, a miner will have to solve a mathematical puzzle that can be complicated enough to take up to 10 minutes with higher computational power. Such a puzzle is unique for that particular block with similar difficulty and miner has to solve it to add that particular block to the blockchain.

Step 5: The miner who succeeded in finding the solution for the mathematical puzzle first, broadcasts this solution to all nodes or miners. This means that he is sending his proof of work to everyone in the distributed network.

Step 6: Then all the miner checks whether the solution resembles the problem of senders block. If it resembles with senders block then miners will grant permission to add that block to the blockchain. Before granting this permission they will again verify the validity of all the transactions. The process in which the majority of the miners agree for a particular transaction, then we can say that consensus is reached.

Step 7: New block always gets added to the top of the blockchain. There might be a possibility of getting multiple 'confirmations' for any block. After the addition of block to the blockchain, the transaction from A to B is updated and B receives money from A. For the addition of a block it will take nearly 10 minutes. After that, all the miners need to start all over again from step three by generating a new block of transactions.

Miners cannot continue the mining old block because of the two reasons. First, it may contain transactions from the last block which are added in the blockchain and therefore they are invalid by making the whole block invalid. Second, every block requires a previous block's hash but miner will not able to add the recently added top block of the blockchain. So, if miner goes the old way, his block will always get rejected.

From the above knowledge, it is observed that blockchain technology is an emerging technology very rapidly. It is a boon for trust issue problems.

So in this thesis, we have proposed an idea to solve the trust issue of software building by using standardized components via blockchain.
In the next chapter, we will be discussing how we can provide trust and quality to software and also the whole process of the composition of software using components.

Chapter 3

Proposed Work

3.1 Introduction

In this chapter, we are going to discuss the whole process of the composition of software using components and a novel approach to provide trust and quality to the software by blockchain technology.

3.2 Proposed Scheme

In component-based software composition, component development and software development by using components are two different aspects done by different developers. There is a need for standardized components for building certified and quality assured software by using plug and play components. Blockchain technology is a better option for standardizing our composite software by quality assured components because of its properties like immutability, non-repudiation, authenticity, integrity and transparency.

This proposed work is a novel approach to certify the components for building software. In this method, blocks of blockchain will be private and open i.e. developer of a

component will be authenticated and components will be used by any third party for constructing software. The core idea is to make every component as a block of blockchain in the peer-to-peer network because of this every component will become immutable, trustworthy and certified.

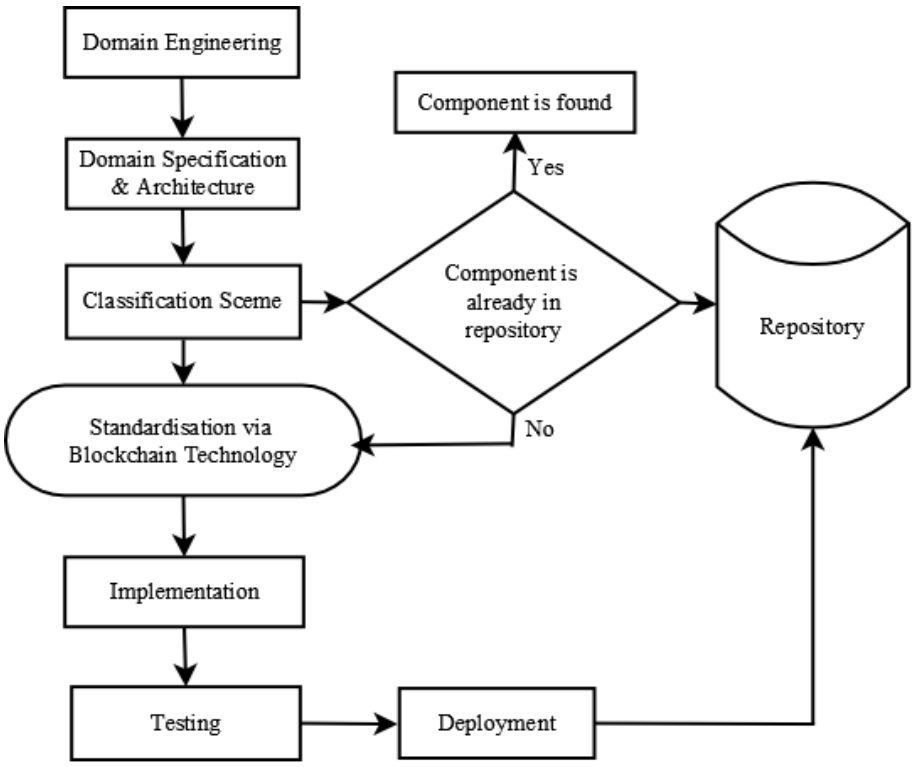

Figure 3.1: Component Development Cycle via Blockchain Technology

All steps of this figure are illustrated below:

3.2.1 Domain Engineering

Domain engineering is also known as product line engineering. It is done to make adaptable and reusable generic components. The development of reusable software is the main concern of domain engineering.

3.2.2 Domain specification and architecture

It helps to make a blueprint, design and framework of components. The component specification describes the functionality, behavior, interaction potential and quality properties of the component.

3.2.3 Classification scheme

Storing new components to the component repository and searching for components for building software, there is a need for classification techniques. In this thesis, an integrated classification scheme is used where the attribute value algorithm and faceted classification scheme are jointly working [15]. Data attributes that are described in the implementation process phase of a block are used in this classification technique which will help to retrieve the components from the repository. By using attribute value and faceted classification schemes, firstly check the repository. If the component is found in the repository then abort the process otherwise move to the implementation process.

3.2.4 Implementation process

In this step, we have to implement our design of components as well as have to standardized our components via blockchain technology. As our approach is to make private and open blocks, every developer of the component should be registered.

The whole process of standardization of components via blockchain technology is described below in a detailed manner:

- *Create a block of blockchain by adding components*

we start this by Initializing genesis block and then create a block with index, timestamp, data, and previous hash. The JSON format is used to store the data. The data part contains attributes details of the component which are operating system, language, function, inputs, outputs, domain, version, and file code of module/component. The hash code of the file in which, the code of component is written, is generated by using InterPlanetary File System (IPFS) [30].

IPFS is a protocol and peer-to-peer network which is used for storing and sharing data in a distributed file system manner. As we know, SHA-256 is a one-way cryptography method that means we can generate the hash code of a file but the reverse is not possible. IPFS is a system that provides the whole file if we know the hash code. The hash code is globally unique in SHA-256 as well as in IPFS, so a third party can get the file after accessing the hash of the block by a distributed file system.

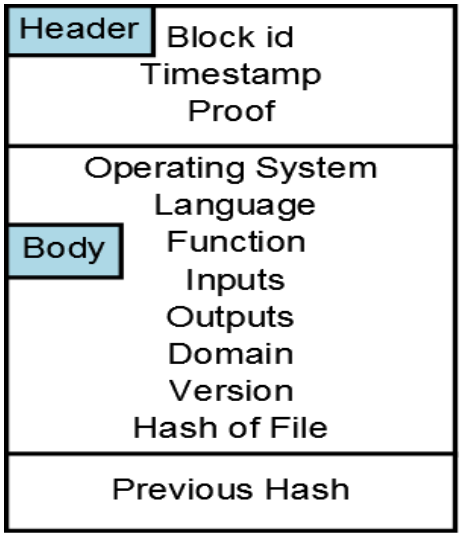

Figure 3.2: Structure of a Block

- *Add digital fingerprint to block*

Cryptographic hash functions are used to prevent and detect any kind of tampering of the data which is stored inside the block. I have used SHA-256 hash function for this purpose because of the avalanche effect, deterministic and one-way property.

- *Implement Proof of Work algorithm*

Proof of Work is an algorithm by which we solve the problem to get the target value by trying different values of the nonce to get a hash that will meet our difficulty criteria.

- *Mining*

Block is picked by miners to calculate the proof of work by solving the mathematical puzzle. Mining is a process to verify a new block. After verification, the new block is added to the blockchain of that node and distributed over a peer-to-peer network.

- *Create interface for user*

Flask is a well known Python micro-framework that is used to create REST API that interacts with distributed users and useful to run several operations in our blockchain node.

- *Establishing Consensus algorithm and decentralization*

In the previous steps, we can see that we are linking block with cryptographic hashes and applying the proof of work algorithm, yet we still can not trust on a single node. We need multiple distributed nodes to avoid single failure and central authority. We are using Distributed nodes to maintain the blockchain. However, there is a problem with these multiple nodes. Due to intentional manipulation, attacks or unintentional reasons like network latency, the copy of blockchain for a few nodes can vary. A consensus algorithm is used to agree upon the longest valid chain for making synchronization over the distributed network.

- *Run the application on postman*

Postman is an Http client, by which we will interact with our blockchain. It is a user-friendly interface where we can access our blockchain operations by Get and Post methods.

The application is made by the following code:

app.run(host = '0.0.0.0', port = port_no)

3.2.5 Verification and testing

This testing is different from the traditional way (figure 2.2) verification and testing. In the traditional method, unit testing, black box testing and white box testing are used. But in this paper, the components are already verified and validated via blockchain technology to follow standardization. So in this step, it only evaluates whether the components are satisfying the need for a new system or not.

3.2.6 Deploy for reuse

In this phase, reusable components are stored in the repository to be used in different software.When a software developer demands components from the repository for building software, he/she will be assured with the quality of component as it is standardized with blockchain technology.

At the time, a software developer builds software by assembling the components, then it looks into the repository for components. If the component is found, then by using the life cycle of figure 2.2, constructs the software. If the component is not found in the repository then by using the life cycle of figure 3.1, firstly develops the component, then stores it into the repository so that in future it can be reused by the third-party software developer.

In this chapter, we discussed the whole process of developing the certified components and also the whole life cycle of constructing the software by reusable trustworthy components. This naive approach standardizes the traditional method of software development by using components. Now in the next chapter, we will discuss the implementation and results for the above approach.

Chapter 4

Implementation and Results

4.1 Introduction

In this chapter, the complete implementation process of standardzing each component and results are described which are presented in preceeding chapters.

4.2 Implementation

The steps of implementation of standardzing each component via blockchain technology are depicted below:

4.2.1 Generate hash code of component file

By using IPFS, will generate the SHA-256 hash code of every component file which are written from scratch by component developer. This hash code will later used by software developer to get the code file of component for building software in distributed manner.

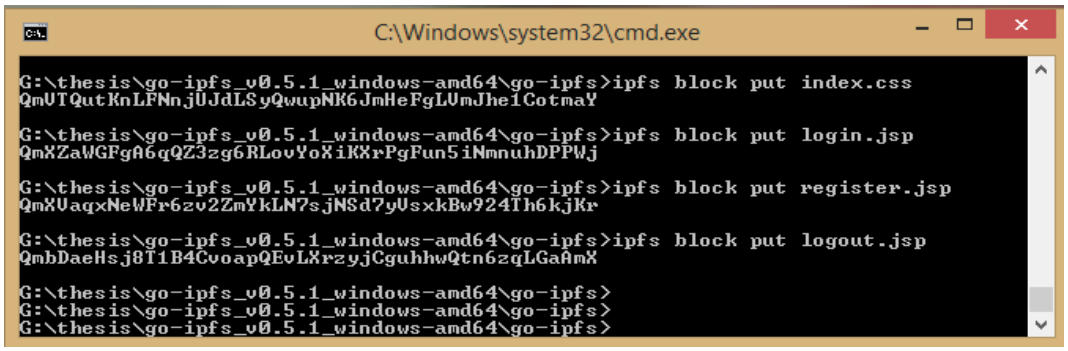

Figure 4.1: Hash Code of Component

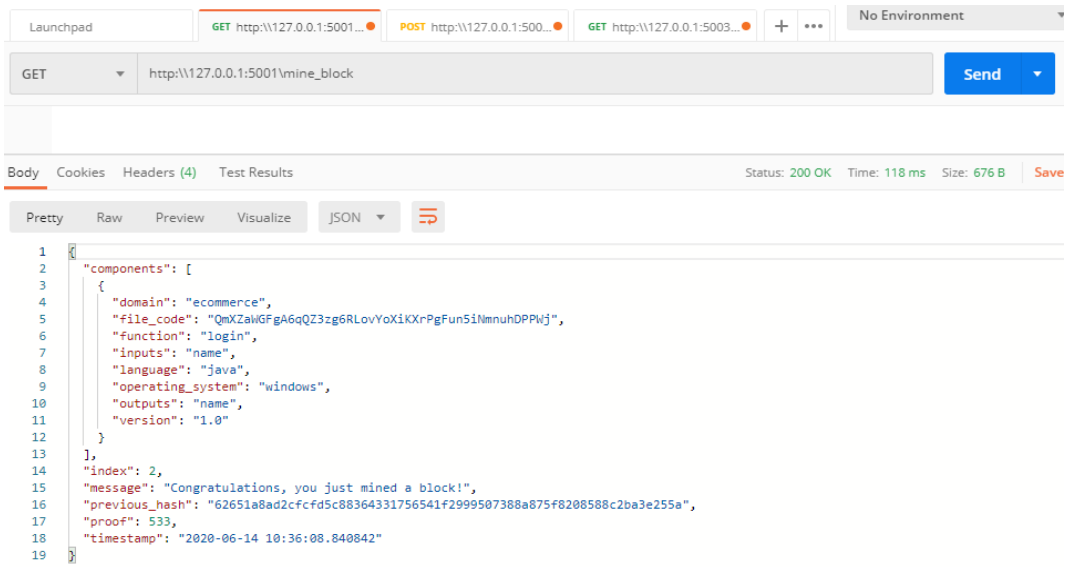

Figure 4.2: Blockchain Implementation of Component

4.2.2 Create blockchain

In this thesis, For creating a blockchain, python language is used. We are using three nodes for a peer-to-peer network which are working on port no. 5001, 5002 and 5003. By using postman, we are going to interact with our blockchain and will perform all the operations of blockchain.

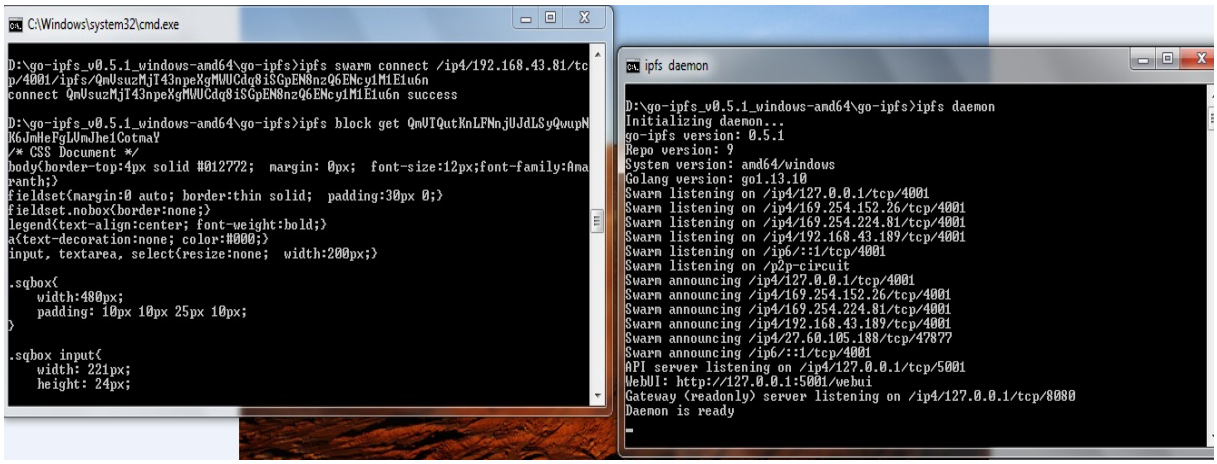

Figure 4.3: Component Code

4.2.3 Retrieve component

When a third party needs to develop software, it looks in repository and select the block on the basis of classification algorithm from blockchain. After getting the hash code from block, the code of file written by component developer is retrieve with the help of IPFS.

4.3 Result

As per requirement, the certification and quality assurance are provided to components for building software. By this whole approach, we can construct the software in place of build the software.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

Blockchain, the same as the Internet, is an open, global infrastructure. It allows companies and individuals for making transactions without any central authority. It reduces the cost of transactions and the time of working through third parties. This technology is based on a distributed ledger and consensus algorithm. This technology allows a digital ledger of transactions to be created and shared among distributed computers on a network. This transaction can be of any application. In my thesis, I have taken it as attributes of the component which will be used for the classification process. The ledger is not owned or controlled by a single central authority or company, it can be viewed and agreed by all users on the network by using a consensus algorithm.

In this thesis, we have focused on certification and standardization of components. For the construction of software in place of development, there is a need for standardization and quality assured components. Quality assurance is performed by black-box testing and white box testing, but it is not so much effective. Blockchain technology provides transparency, trust, reliability and security. For making trustworthy components and giving

standard to CBSE, blockchain technology is a very efficient approach.

In my thesis work, I have created components as a block, linked them by using SHA-256 cryptography and distributed them over a decentralized network. For selecting and classifying the components, faceted based algorithm and attribute value based algorithm are used. Attributes and hash form of file code of every component is stored in the data part of the block so that in future no one can alter the code of components. Hash code of component can set and get by using IPFS.

5.2 Future Direction

As we know, blocks of blockchain are immutable. It is not possible to change a single character in the block because of the avalanche effect of SHA-256. So, there will be a problem with updating components in the future. Apart from this, in the future can propose a new consensus algorithm for validating the block.

References

[1] Sajjad Mahmood, Richard Lai, Y. S. Kim: Survey of component-based software development. IET Software 1(2): 57-66 (2007)

[2] William B. Frakes, Kyo Kang: Software Reuse Research: Status and Future. IEEE Trans. Software Eng. 31(7): 529-536 (2005)

[3] Tassio Vale, Ivica Crnkovic, Eduardo Santana de Almeida, Paulo Anselmo da Mota Silveira Neto, Yguaratã Cerqueira Cavalcanti, Silvio Romero de Lemos Meira: Twenty-eight years of component-based software engineering. Journal of Systems and Software 111: 128-148 (2016)

[4] Xia Cai, Michael R. Lyu, Kam-Fai Wong, Roy Ko: Component-based software engineering: technologies, development frameworks, and quality assurance schemes. APSEC 2000: 372-

[5] Bawa, R., and Iqbaldeep Kaur. "Algorithmic Approach for Efficient Retrieval of Component Repositories in Component based Software Engineering." Indian Journal of Science and Technology 9 (2016): 27-70.

[6] KungKiuLau, ZhengWang: A Taxonomy of Software Component Models. EUROMICRO SEAA 2005: 88-95

[7] Khan, Asif Irshad, and U. A. Khan. ”An improved model for component based software development.” Software Engineering 2.4 (2012): 138-146.

[8] Crnkovic, Ivica. ”Component-based software engineering—new challenges in software development.” Software Focus 2.4 (2001): 127-133.

[9] Aoyama, Mikio. ”New age of software development: How component-based software engineering changes the way of software development.” 1998 International Workshop on CBSE. 1998.

[10] Bassam, A. ”Reusable Software Component Life Cycle/Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar, et. al–NAUN.” International Journal of Computers, ussue 2: 191-199.

[11] Antovski, Ljupcho, and Florinda Imeri. ”Review of Software Reuse Processes.” International Journal of Computer Science Issues (IJCSI) 10.6 (2013): 83.

[12] Vitharana, Padmal. ”Risks and challenges of component-based software development.” Communications of the ACM 46.8 (2003): 67-72.

[13] Wang, Ju An. ”Towards component-based software engineering.” Journal of Computing Sciences in Colleges. Vol. 16. No.1. Consortium for Computing Sciences in Colleges, 2000.

[14] Kaur, Vaneet, and Shivani Goel. ”Facets of software component repository.” International Journal on Computer Science and Engineering 3.6 (2011): 2473-2476.

[15] Rao, C. V. G., and P. Niranjan. ”A Mock up tool for Software Component reuse repository.” International Journal of Software Engineering and Applications 1.2 (2010).

[16] Jeng, Jun-Jang, and Betty HC Cheng. ”Using formal methods to construct a software component library.” European Software Engineering Conference. Springer, Berlin, Heidelberg, 1993.

[17] Fischer, Bernd. ”Specification-based browsing of software component libraries.” Automated Software Engineering 7.2 (2000): 179-200.

[18] Soora, Sathish Kumar. ”Feature Based Classification Retrieval of Reusable Components.” Int. J. on Advanced Computer Theory and Engineering (IJACTE) 1 (2012).

[19] Verma, Amit, Monisha Kumari, and Iqbaldeep Kaur. ”Software Component Retrieval using Keyword-Based Search Technique.” International Journal of Computer Science and Information Security 15.1 (2017): 105.

[20] Moritz Beller, Joseph Hejderup: Blockchain-based software engineering. ICSE (NIER) 2019: 53-56

[21] Nakamoto, Satoshi. ”Bitcoin: A peer-to-peer electronic cash system.”(2008).

[22] Li, Xiaoqi, et al. ”A survey on the security of blockchain systems.” Future Generation Computer Systems (2017).

[23] Guang Chen, Bing Xu, Manli Lu, Nian-Shing Chen: Exploring blockchain technology and its potential applications for education. Smart Learn. Environ. 5(1): 1 (2018).

[24] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, C. Yang, The blockchain as a decentralized security framework [future directions], IEEE Cons. Electr. Mag. 7 (2) (2018) 18_21.

[25] Ulybyshev, Denis, et al. ”(WIP) blockhub: Blockchain-based software development system for untrusted environments.” 2018 IEEE 11th International Conference on Cloud Computing (CLOUD). IEEE, 2018.

[26] Nizamuddin, Nishara, Haya R. Hasan, and Khaled Salah. ”IPFS-blockchain-based authenticity of online publications.” International Conference on Blockchain. Springer, Cham, 2018.

[27] Hanifatunnisa, Rifa, and Budi Rahardjo. ”Blockchain based e-voting recording system design.” 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA). IEEE, 2017.

[28] Yi, Haibo. ”Securing e-voting based on blockchain in P2P network.” EURASIP Journal on Wireless Communications and Networking 2019.1 (2019): 1-9.

[29] Saberi, Sara, et al. ”Blockchain technology and its relationships to sustainable supply chain management.” International Journal of Production Research 57.7 (2019): 2117-2135.

[30] Kumar, Randhir, and Rakesh Tripathi. ”Blockchain-Based Framework for Data Storage in Peer-to-Peer Scheme Using Interplanetary File System.” Handbook of Research on Blockchain Technology. Academic Press, 2020. 35-59.