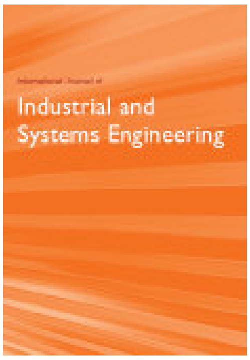


International Journal of Industrial and Systems Engineering

ISSN online: 1748-5045 - ISSN print: 1748-5037
https://www.inderscience.com/ijise

NBSOC framework for team structure to develop blockchain-based applications

Nitish Joshi, Tejaswi Khanna, Vikram Bali, Shivani Bali

DOI: 10.1504/IJISE.2022.10049488

Article History:

Received:	25 July 2021
Accepted:	10 May 2022
Published online:	08 January 2024

NBSOC framework for team structure to develop blockchain-based applications

Nitish Joshi

Oracle Tech Hub,
Marathahalli – Sarjapur Outer Ring Rd, Kadubeesanahalli,
Bengaluru, Karnataka 560103, India
Email: nitishjoshi3@gmail.com

Tejaswi Khanna

Computer Science and Engineering Department,
Sharda University,
Greater Noida, Uttar Pradesh, India
and
Tower 20, 1404, Shri Radha Sky Garden,
Sec. 16B, Greater Noida West – 201308, India
Email: Tejaswi.khanna@gmail.com

Vikram Bali*

IMS Engineering College,
NH-09, Adhyatmik Nagar, Near Dasna,
Distt: Ghaziabad, Uttar Pradesh – 201015, India
Email: vikramgcet@gmail.com
*Corresponding author

Shivani Bali

Jaipuria Institute of Management,
A-32A, Sector-62, Noida, Uttar Pradesh – 201309, India
Email: lbsshivani@gmail.com

Abstract: For the development of blockchain applications, platforms have been implemented by corporates like IBM, Oracle, Amazon, etc. Blockchain comprises smart contract development, which gets deployed over a peer-to-peer network. Basic skills about blockchain are still lacking amongst the developer community and all the people involved in developing blockchain applications. This paper proposes an NBSOC framework for organising teams to build blockchain-based applications. This framework has been used to create a team structure for implementing a land record management system. The authors have addressed the implementation challenges, cost, roles and responsibilities of an individual in the blockchain development environment.

Keywords: blockchain; team structure; smart contract; decentralised applications; software engineering.

Reference to this paper should be made as follows: Joshi, N., Khanna, T., Bali, V. and Bali, S. (2024) 'NBSOC framework for team structure to develop blockchain-based applications', *Int. J. Industrial and Systems Engineering*, Vol. 46, No. 1, pp.126–149.

Biographical notes: Nitish Joshi is a Software Professional who has previously worked with companies like Deloitte and Accenture. He has been awarded for working on Cyber EYE project funded by Department of Science and Technology, Government of India. Currently, he is serving as a Solution Engineer in the Oracle India Pvt. Ltd. His research interest includes blockchain, network, software modularity.

Tejaswi Khanna is a PhD scholar in the Sharda University, Greater Noida. Currently, he is serving as an Assistant Professor in the Computer Science and Engineering Department in the same varsity, he is a member of IAENG and IRED. His research interest includes computer networks, database management systems, distributed system, blockchain, network security and social networking.

Vikram Bali is the Director, IMS Engineering College, Ghaziabad, Uttar Pradesh, and has more than 22 years of teaching and administrative experience. He had graduated from REC, Kurukshetra – BTech (CSE), post-graduation from NITTTR, Chandigarh – M.E (CSE) and Doctor of Philosophy (PhD) from Banasthali Vidyapith, Rajasthan. He has published more than 50 research papers in international journals/conferences, authored five text books and six patents. He is a Series Editor for three book series of CRC Press – Taylor & Francis Group. He is a lifetime member of IEEE, ISTE, CSI and IE. He has completed two sponsored research projects, funded by TEQIP-III and Unnat Bharat Abhiyaan by MHRD, GoI.

Shivani Bali is an academician, trainer, researcher and consultant with more than 17 years of experience in the field of decision sciences and operations management. She is working as a Professor at the Jaipuria Institute of Management, Noida. She has published around 30 research papers indexed in SCI/WoS/ Scopus journals and published four patents. She has been working on consulting assignments with many reputed companies and has imparted training programs in the area of data science and business analytics. Her areas of interest are business analytics, data science, operations and supply chain analytics.

1 Introduction

Blockchain is a term that has been tossed quite loosely around many times since 2008 when Nakamoto (2008) released the paper on bitcoin. In the realm of computer science and systems, but finance and banking, governance, and healthcare, blockchain is being viewed as a disruptive technology that can enhance the working of these industries for the better. Deriving benefits from already established technologies of networking, cryptography, and distributed systems, blockchain can be viewed as a distributed ledger technology disrupting how business processes function.

The academic and industrial literature is populated with comprehensive views of the concepts, facts, and features of the technology and some have successfully implemented a handful of use cases. Yli-Huumo et al. (2016) has observed that over 80% of the literature is focused on Bitcoin, and less than 20% deals with other Blockchain applications. In their bibliometric analysis of blockchain literature of 995 papers from 2013 to 2018, Dabbagh et al. (2019) have similar observations that the research interest has shifted from bitcoin to blockchain. A systematic literature review presented by Andrian et al. (2018) divides the technology into three components, smart contract, distributed system, and cryptography. Hence, it is imperative to clarify the technological and human resource organisational structures required to develop blockchain applications.

Knowing the facts and the features of the technology may quickly help business owners to identify blockchain as the potential technology to increase its transactions strength and reliability. Still, it is also needed to determine the human resource team which can enable them to implement it successfully on their systems. What essentially lacks in almost all of the literature is an organisational structure to execute the application. In other words, the question of *who develops what* needs to be answered. For this, the authors have proposed a model for creating a team structure to design, develop and deploy a blockchain-based land registry system. In this paper, the authors focus on the implementation requirements, choice of programming languages, development life cycles, integrations, and roles and responsibilities of an individual in the blockchain team.

The primary research questions this paper aims to answer are:

RQ1 What are the human resources or team components required to build a blockchain application?

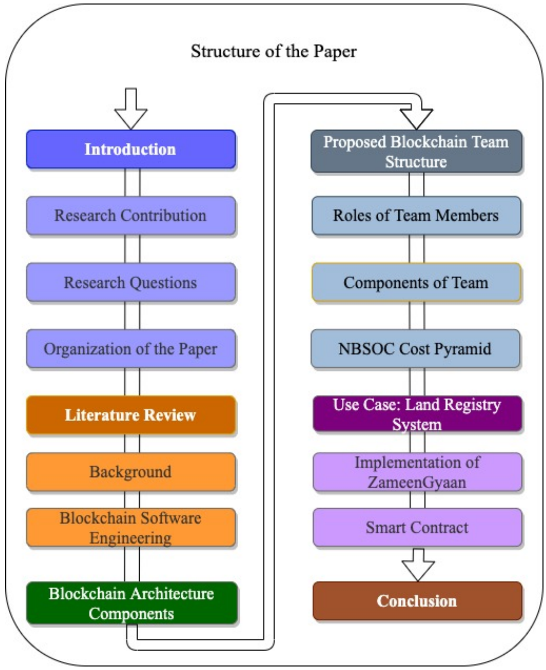
Developing a blockchain application requires a diversified team of professionals in the technology space. One has to look for the expertise and the technology sub-domain needed while creating or recruiting the team. The authors have tried to answer this by presenting a team framework to measure the human resource components for the blockchain application to be built seamlessly.

RQ2 What are the technical or infrastructural components required to build a blockchain-based land registry system?

Land registry system is a powerful application that can be developed over a blockchain network. Just like there needs to be clarity on team recruiting, the developers must decide which infrastructural components have to be involved. What type of services of blockchain can be outsourced? What kind of network security protocols can be used? The authors have implemented a blockchain-based land registry system, understood the requirements, and shared their implementation results to share a little idea.

The paper is organised as follows. Section 1 is the introduction which commences with the research question that this paper aims at resolving. Section 2 gives a detailed background about blockchain and its already proposed implementations, proof of concepts, or solutions. Section 3 is the literature review which talks about taxonomy and blockchain-based software engineering. Section 4 presents the essential blockchain components required for the answer in this paper. Section 5 proposes the team structure for blockchain application development. Section 6 discusses the use case being taken into consideration for releasing the proposed methodology. Then the implementation is showcased in Section 7 with finally ending the paper with the conclusion section. Figure 1 shows the structure of the article as discussed.

Figure 1 Structure of the paper (see online version for colours)



2 Literature review

The following section will discuss the previous literature studied to provide a comprehensive view of the gaps. It houses background as the starting subsection, which presents a perspective on introducing the technology and its potential in developing enterprise-grade systems. Then the authors give a view on what they call blockchain software engineering. This subsection presents a concise statement of the plethora of research and proposals coinciding with blockchain with software engineering. Table 1 gives a brief overview of the literature reviewed in blockchain software engineering.

2.1 Background

Blockchain is an innovation in information technology that uses distributed ledger technology to validate transactions without a trusted third party (TTP). It is achieved by employing what is called a consensus algorithm. The consensus algorithm plays a vital role in maintaining the safety and integrity of the transactions. Zibin et al. (2018) observed a gap between the technical and the application aspects of the said technology, where most of the research focuses primarily on applications. They have conducted a comprehensive survey describing blockchain taxonomy and technical challenges being faced while developing blockchain solutions. Mingxiao et al. (2017) and Zheng et al. (2017, 2018) have put forth literature reviews enlightening about the various consensus algorithms being implemented in the blockchain.

Table 1 Bibliographic overview of the literature surveys

S. no.	*Author details*	*Year of publication*	*Publication type*	*Publisher/journal/ proceedings*	*Contributions*	*Study design*	*Challenging area*
1	Xu et al.	2017	Conference	IEEE	Proposed a taxonomy to design and evaluation of software architectures using blockchain	Taxonomy design	Data privacy and scalability
2	Dinh et al.	2017	Conference	Association for Computing Machinery	Proposed a framework to analyse the performance of private blockchains	Performance analysis	Scalability and performance
3	Pongnumkul et al.	2017	Conference	IEEE	A methodology for evaluating a blockchain platform is developed	Performance analysis	Performance
4	Rouhani and Deters	2017	Conference	IEEE	Studied ethereum transactions and analyse ethereum clients, Geth and Parity, on a private blockchain.	Performance analysis	Scalability and performance
5	Porru et al.	2017	Conference	IEEE	Presented the need of blockchain oriented software engineering	Software design	Tools for smart contract development and testing are required
6	Liao et al.	2017	Conference	IEEE	Presented a service platform that supports behaviour-driven development (BDD)-style for smart contract	Framework design	Smart contract programming issues
7	Tasca and Tessone	2018		SSRN	Proposed taxonomy of blockchain application	Taxonomy design	Security and privacy
8	Anascavage and Davis	2018		SSRN	Concluded that blockchain research is still theoretical and not applied	Review study	Blockchain implementation
9	Marchesi et al.	2018	Conference	Association for Computing Machinery	Agile software engineering method for blockchain application design	Software design	Blockchain implementation
10	Rocha and Ducasse	2018	Conference	IEEE/ACM	Steps for blockchain modelling notation	Software modelling	Blockchain implementation
11	Chakraborty et al.	2018	Conference	IEEE/ACM	Study of conventional software engineering on blockchain applications	Software design/case study	Implementation
12	Perboli et al.	2018	Journal	IEEE	Studied the role of blockchain in reducing logistics cost and in optimising operations	Case study, cost analysis	Integration of blockchain with business process modelling
13	Wessling et al.	2018	Conference	IEEE	Proposed architectural patterns for blockchain-based applications	Software design	Models, tools, and methods for blockchain-oriented software engineering are not yet developed

Table 1 Bibliographic overview of the literature surveys (continued)

S. no.	*Author details*	*Year of publication*	*Publication type*	*Publisher/journal/ proceedings*	*Contributions*	*Study design*	*Challenging area*
14	Destefanis et al.	2018	Conference	IEEE	Presented the need of blockchain oriented software engineering, best practices to develop smart contracts	Smart contract analysis	Smart contract programming issues
15	Tikhomirov et al.	2018	Conference	IEEE	The design of smart contracts have been analysed	Static analysis	Smart contract bug detection
16	Hegedus	2018	Conference	IEEE	Analysis tools and studies for smart contracts	Contract analysis using a prototype tool	Smart contract development and deployment issues
17	Pradeep et al.	2018	Conference	IEEE	Proposed a model for automated analysis; identify the digitisability level of the given regulations to smart contracts	System modelling	Smart contract development
18	Sukhwani et al.	2018	Conference	IEEE	Presented a performance model of Hyperledger Fabric	Performance analysis	Ordering services and ledger writing
19	Koul	2018	Conference	IEEE	Highlights the challenges currently faced in testing such applications	Review study	Blockchain software testing
20	Seebacher and Schrutz	2019	Conference	European Conference on Information Systems	Implementation challenges from conventional information systems	Comparative study	Organisational and network challenges
21	Bosu et al.	2019	Journal	Empirical Software Engineering	Studied the needs of blockchain software developers	Review study	Roles of team members, smart contract debugging, and testing tools
22	Vishwanathan	2019	Journal	IBM	Blockchain solution reference architecture	Software design	Blockchain software development
23	Sillaber et al.	2020	Journal	Information Systems and e-Business Management.	Smart contract development	Software design, team roles	Smart contract design
24	Hajdu et al.	2020		Arxiv	Tested faulty smart contracts and blockchain network behaviour using software-implemented fault injection (SWIFI)	Performance analysis	Blockchain software testing, blockchain fault tolerance
25	Vingerhouts et al.	2020	Conference	CEUR Workshop Proceedings	Provide modelling frameworks to represent supply chain processes in blockchain	Software design	Mapping of supply chain processes in blockchain
26	Current work	-	-	-	Blockchain team structure provided technical details about deployment. Blockchain software cost model proposed.	Blockchain software engineering	Blockchain for real estate and land registry system.

Puthal et al. (2018) and Adhi Tama et al. (2017) have presented various application scenarios which can successfully implement blockchain. Using the framework proposed by Wüst and Gervais (2018), development teams can comprehend the need for blockchain technology for a particular business or government application. Not only that, business processes can be optimally executed on top of commodity blockchain technology, as presented by García-Bañuelos et al. (2017). Ahram et al. (2017) have shown the various benefits that blockchain garners to multiple industries apart from finance by developing a healthcare industry application.

Before developing any blockchain-based application, it is imperative to identify blockchain 1.0 or 2.0 or 3.0 or 4.0. Swan (2015) provides insights into blockchain being a blueprint for the new economic and governance systems. As a paradigm shift in businesses, Kizildag et al. (2019) present blockchain application in contemporary hospitality management. Tsiulin et al. (2020) reviewed blockchain applications in shipping and port management while defining three main conceptual areas: document workflow management, financial processes, and device connectivity.

Apart from its potential to change the business application scenario by reducing mediators for validating transactions, it is also being used successfully as a supporting platform for the internet of things as observed by Yu et al. (2018) and AI development with the ownership preservation protocol presented by Somy et al. (2019). By imparting trust as a systemic feature, researchers have explored the use of blockchain in implementing traceability in supply chains, as presented by Malik et al. (2019) and Khanna et al. (2020). Forwarding the trust management that it provides, blockchain can be used to securely access electronic health records (EHR) as proposed and evaluated by Guo et al. (2019), Vora et al. (2018), Bali et al. (2021) and Tanwar et al. (2020).

When building blockchain applications, the developers have to understand and utilise the property of decentralisation. This feature is well used by Baza et al. (2019) and Mylrea and Gourisetti (2017) for coordinating smart grid energy units. Another issue to be addressed is whether the blockchain ledger has to be stored and managed via a blockchain provider (like Ethereuem or Hyperledger) or on-premise. Consequently, blockchain-based architectures have been proposed for industrial IoT for backing hierarchical storage (Wang et al., 2019) and scalable reconfiguration (Le-Dang and Le-Ngoc, 2019). They have to create a mechanism to handle the delays in executing the smart contract transaction calls. Furthermore, decisions regarding how users interact with their applications have to be considered by the development team – categorising different architectural patterns of decentralised applications (DApps), Wessling and Gruhn (2018) hint towards the importance of considering how the user interacts with the decentralised apps.

2.2 Blockchain software engineering

This paper lies in the scope of Blockchain Software Engineering, which is different from blockchain-based software engineering as presented by Beller and Hejderup (2019), where they have viewed blockchain technology to be a solution to the problem of continuous integration (CI) in software engineering. However, this paper builds upon the existing taxonomy to present a preferred team structure for developing blockchain applications; a cost function is also shown, enabling software managers and solution providers to analyse the cost of development and deployment of blockchain-based software. Singh et al. (2022) proposed an optimisation of intrusion detection systems determined by ameliorated HNADAM-SGD algorithm to provide the security.

Xu et al. (2017) have proposed a taxonomy to classify and compare different blockchains. According to them, ‘blockchain has become a publicly-available infrastructure for building decentralised applications and achieving interoperability’. The taxonomy can be used when comparing blockchains and designing and evaluating software architectures using blockchain technology. The basic properties of blockchains are immutability, non-repudiation, data integrity, data transparency, and equal rights. However, data privacy and scalability are non-functional properties that are points of criticism. In a comparative study presented by Seebacher and Schüritz (2019), it has been observed that learning the implementation challenges from conventional information systems can be applied for blockchain applications. These include but are not limited to technical, organisational, and network challenges.

2.3 Blockchain architectures and taxonomy

The architecture design to implement decentralisation in the network, applications can be partially centralised and partially decentralised. For this, two options are to be studied, permission and verification. Permission involves permissioned blockchains where the participants have to have permission to join and validate the transactions and, in turn, mine the blocks. In terms of verification, there can be a central or distributed verifier (Oracle) to evaluate the state and validate transactions and smart contracts accordingly. Bitcoin employs the mining mechanism for verification, and hyper ledger uses orderers for performing the same task. Not only Bitcoin, but the blockchain is also rapidly evolving into different blockchain technologies.

Referred to as heterogeneity by Tasca and Tessone (2018), this evolution into different verticals might pose significant concerns. Solving this issue, they have proposed a taxonomy of blockchain applications by disentangling them into individual building blocks. The main components of any blockchain application are observed as:

- consensus
- transaction capabilities
- native currency/tokenisation
- extensibility
- security and privacy
- codebase
- identity management
- charging and rewarding system.

Now, blockchain as a data structure stores a lot of data about the transactions, and the blocks are formed because of computational power, which costs money, and lack of it affects performance. Regarding cost efficiency, performance and flexibility, there comes a question regarding what data needs to be stored on-chain and what has to be stored off-chain.

While deciding the team structure, use cases for blockchain development also play a role in determining the functions of the individuals. Is a fantastic fact worthy of being brought to the readers' attention is that the review of Anascavage and Davis (2018) has concluded with a section specifically titled *how*? Most of the blockchain research is still theoretical and not applied. Blockchain can be implemented in an integrated and sizeable complicated system, but the clarity is still further researched. Keeping basic software engineering concepts in mind, Marchesi et al. (2018) propose a software development process following the conventional SDLC phases. Namely, the requirement, analysis, design, development, test, and deployment. Rocha and Ducasse (2018) have proposed preliminary steps as specialised blockchain modelling notations.

2.4 Blockchain software engineering vs. traditional software engineering

Since ledger and smart contract code are deployed over decentralised systems, standard software engineering methods including testing and security best practices should be brought in alignment with more seriousness in the interest of mitigating potential threats. After analysing 156 responses, Chakraborty et al. (2018) observed that the requirement collection of blockchain applications is different from conventional software. However, the developer team can decide on work assignments voluntarily. In their research, Bosu et al. (2019) concluded that software development tools for non-blockchain software are insufficient for blockchain development. There is a requirement of customised IDEs, smart contract debugging and testing devices, and blockchain-specific design notations and deployment simulators.

A generic framework has been proposed by Sillaber et al. (2020) to support application development involving distributed ledgers. Their research focused on smart contract development and integration. They have divided the whole application development process into three major elements, vis. roles, activities, and artefacts. Roles can encompass innovative contract engineers, software engineers, and legal experts. All of these are responsible for the outcome of their actions. These outcomes comprise either creation or updation of artefacts which can be referred to as the input and output of individual or team activities. Artefacts are created, modified, and used during the transaction and are either the final product, parts of it, or intermediate results. Examples of artefacts include concepts, models, source code, smart contract code, or documents such as performance reports.

2.5 Blockchain software analysis

To date, analysis of blockchain involves security and network analysis of the blockchain system as a whole. Dinh et al. (2017) have proposed Blockbench, a framework to analyse the performance of private blockchains in terms of throughput, latency, scalability, and fault-tolerance. More research has been carried out to analyse the performance of ethereum blockchain applications rather than hyper ledger projects. Pongnumkul et al. (2017) presented their observation that on very high workload scenarios, the performance of private blockchains can not be considered competitive with the current in-use database systems. Their analysis, however, was conducted on both ethereum and hyper ledger fabric, where they have shown that fabric outperforms ethereum in all the considered parameters. Another evaluation of ethereum clients, parity, and geth was carried out by

Rouhani and Deters (2017), showing that the transactions are 89.8% on average faster in parity.

There is a specific lack of standard methodology for designing a strategy that can be utilised to develop and validate the overall Blockchain solution and further integrate it into the business strategy. Employing GUEST strategy for use case design, Perboli et al. (2018) have established that there is a need to define the different actors involved along with their jobs, the gains (benefits), and the pains (problems).

Wessling et al. (2018) have also provided a more fine-grained approach to decide which elements of an application architecture could benefit from blockchain technology. Identifying participants, their trust relations, and interactions derive a hybrid architecture from using blockchain only in certain parts of the software system. With the advent of smart contract execution provided by Ethereum, Porru et al. (2017), and Destefanis et al. (2018) have even called for the definition of blockchain software engineering with a view of creating standardised best practices of smart contract programming and testing. Liao et al. (2017) observed that verification of smart contract logics mapping with business requirements is an underrated software engineering issue in blockchain application development. Utilising behaviour driven development, they proposed a platform that could make smart contract development more efficient and less burdensome.

2.6 Blockchain smart contract design

Additionally, the design of smart contracts has been analysed extensively, as presented in their static analysis by Tikhomirov et al. (2018). Using conventional methods like UML class and sequence diagrams defined by Kaur and Bali (2015) to analyse and test contract codes may or may not be feasible. Exploring the complexity of solidity smart contracts, Hegedus (2018) proposed adopting standard object-oriented metrics and applied them to analyse 10,000 smart contracts. The research showed that smart contract development could be more efficient using external libraries and dependency management mechanisms.

Another research was presented by Pradeep et al. (2018), where they have proposed a modelling approach for automated analysis of human-readable regulation representations and equivalent smart contract components. In addition, their model also suggests a suitable blockchain environment required for application development. Sukhwani et al. (2018) have evaluated permissioned blockchain using stochastic reward nets in terms of throughput and utilisation of network resources. They found out that there is a performance bottleneck being created by ordering services and ledger writing.

The development and testing of smart contracts are paramount to the success of the whole blockchain software. Koul (2018) acknowledges a need to establish customised testing tools and processes for blockchain-oriented software. Typical software faults in smart contracts can have huge impacts, as analysed by Hajdu et al. (2020) using software implemented fault injection (SWIFI) technique and found out that formal verifications and runtime protections are needed blockchain systems more dependable.

There remains a gap in defining the team organisation required for developing and implementing blockchain solutions at the enterprise level. It has been established that while developing software-based upon blockchain, there needs to be a network-level configuration and deployment in addition to the smart contract layer, which implements the business logic over the deployed blockchain network. A fantastic reference architecture has been proposed by Vishwanathan et al. (2019), in which they present member onboarding architecture consisting of a set of processes and activities which are required to be performed by the entities of participating organisations to transact and exchange information. However, this occurs after the blockchain solution has been implemented and deployed.

Using farm-to-fork as a case study, Vingerhouts et al. (2020) provide modelling frameworks to represent supply chain processes in the blockchain. They utilise the previous knowledge of i* framework and UML use case and sequence diagrams.

3 Blockchain architecture components

Blockchain consists of nodes, channels, peers, chain codes, Databases, and ledgers that containerise in a virtual machine (VM). VMs can be deployed either on cloud or on-premise server spaces. An overview of all these components is presented as:

- Nodes: all the participants in the blockchain network are called nodes, e.g., RERA, Land Registry Office. The major responsibility of these nodes is to validate and verify the transactions before they can be added to the blockchain ledger. The verification process is managed by predefined consensus algorithm of the blockchain framework employed for deploying the system.
- Channels: all participants can communicate with each other through the media. Channels are separated based on organisations to provide segregation of information exchange.
- Peers: all nodes should have at least two peers (endorsing and committing peers), which help nodes to verify and validate the transactions. These peers are chosen by the administrator during the deployment of the application chain code.
- Chain codes: chain codes, also known as smart contracts, where developers can write CRUD operations that act on world states or databases accordingly. Chain codes are the application logic of the blockchain system which defines who owns the rights to perform which operation on which asset.
- Ledgers: ledgers are responsible for storing the transactions and creating blocks by generating each transaction hash. They are private since the proposed methodology is a private permissioned blockchain. However, certain data points can be made publicly accessible.

Many technical communities are working and approaching developing open-source blockchain platforms that will offer blockchain and their actors. There are two blockchain platforms available currently, private or permissioned blockchain and open or public blockchain. Public blockchain and permissioned blockchain platforms can be used to solve the same purpose, but there are some fundamental differences in their implementation as in:

1 Public blockchain, there is no security on data, which is very important for enterprise applications. Still, permissioned blockchain offers protection on the data by guaranteeing its ownership, as Mitani and Otsuka (2020) described.

2 Public blockchain works on the non-deterministic algorithms to solve the proof of work by using mining techniques that generally impact its performance. In contrast, permissioned blockchain works on deterministic consensus algorithms and can handle vast numbers of transactions per minute.

These two features create a massive difference for both platforms and increase the scope of public and permissioned blockchain cases. Currently, Hyperledger fabric is the permissioned blockchain that is developing under the Hyperledger project of the Linux Foundation. Many leading companies have already deployed cloud services using Hyperledger as the underpinned technology for the blockchain platform. So now, there is a need to understand the team structure of the blockchain technology that will help the organisation manage the blockchain platform and the smart contracts from development to production environment.

4 Proposed blockchain team structure

4.1 Roles of team members

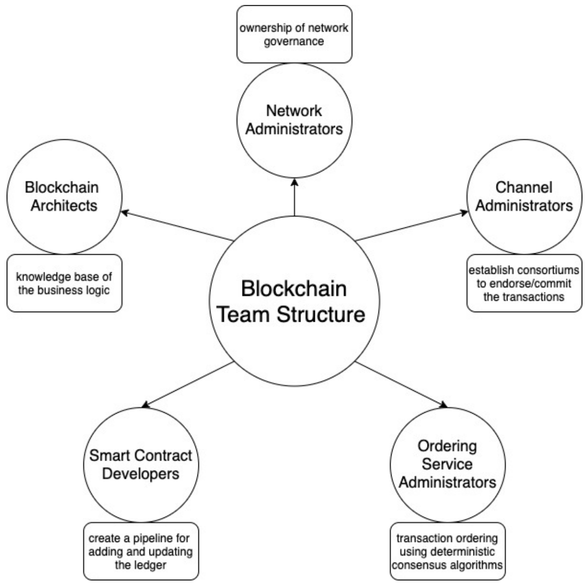
The roles of the team members can be broadly categorised into admin roles and developer roles. Any individual part of the development team can self-assign these roles since the application is distributed in nature. Being largely open-source, blockchain applications can utilise the self-assignment mechanism for task assignment as described by Crowston et al. (2007).

- *Admin role:* blockchain platforms are the space or container where the user records various transactions. New transactions pass through the complex architecture where the platform validates. Consensus algorithms give their consents to the transactions via peers, which are defined in the system. Each node will have access to a copy of the transaction. Setting up peers, channels, and server admins is the responsibility of Blockchain admins, which provide the infrastructure space to the smart contracts. Blockchain admins are also responsible for maintaining the logs and providing enough room for the system to perform efficiently.
- *Developer role:* developers are responsible for writing the chain codes with the invocation of all the libraries required to invoke the necessary functions to create or update the world state of blockchain architecture. Currently, we are free to choose any language among Go, Java, and Node Js as all these languages have concurrent programming features. Preferably most of the programmers are using Go language to code the smart contracts. Developers are only responsible for writing the smart contracts as per the use case and deploying them on the blockchain platform.

4.2 Components of the team

Hyperledger fabric has offered their tutorials documentation for the following set of audiences, so their roles will be covered in this document to understand who the primary owners of specific components will be. Roles, experience, and technical prowess are discussed. Figure 2 represents NBSOC and the different components of the team, along with their significant roles in the application development.

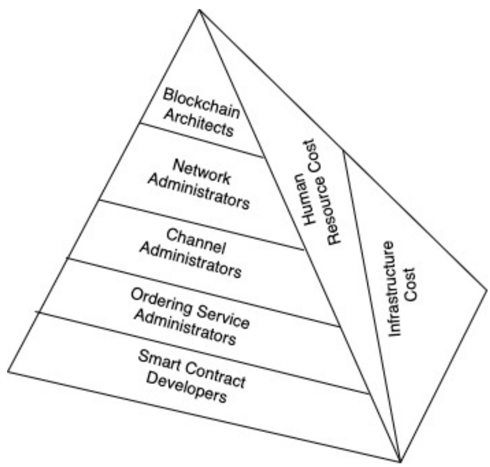
Figure 2 Proposed NBSOC blockchain team structure and roles



- *Architects:* they can also be named blockchain architects who will be the primary owner of the complete solution implemented. They will be the ones who will direct the actors in case of any changes needed to make in the blockchain solution. They are the knowledge base of the business logic of the application. They are aware of the whole technology stack and the design of the application. They are the most experienced personnel in the team. They are good with APIs, communication skills for delegation.
- *Network administrators:* the blockchain should be decentralised in nature. All the participants who want to join the network, either part of the subnets or networking team, should make it easy for them to exchange information by identifying the participants using valid CA. Hence the ownership of network governance is with the network administrators.
- *Channel administrators:* channels are very pivotal components of blockchain networks. They provide the space for the participants to establish consortiums where they can endorse or commit the transactions on the ledgers by their supporting and engaging peers, respectively. There should be a team or an owner who can govern this component by accumulating the knowledge on it for the channels. There can be a case where the number of channels can increase as the number of participant's increases.
- *Ordering service admin:* ordering service is a critical component as it makes the permissioned blockchain different from the public blockchain by ordering the transaction using the deterministic consensus algorithms. Internally, Hyperledger uses Kafka and Raft for implementing ordering mechanisms that can be multimode, and hence it needs special care and ownership by the teams.

- *Application and smart contract developers:* by just reading the documentation of Hyperledger, it seems that it's a distributed ledger technology of immutable transaction logs. But the SDK of blockchain needs a programming language that will create a pipeline for getting and setting the transactions in the ledger. Hyperledger uses shim libraries with abstract methods and functions that help developers develop the code that can interact with the ledgers. These smart contracts will have to be written by organisation developers according to business logic and requirements. Currently, Hyperledger works on Golang, Java, and NodeJS to develop smart contracts.

Figure 3 NBSOC cost pyramid



4.3 NBSOC cost pyramid

Figure 2 represents a proposed cost pyramid from the NBSOC framework, including human resource cost and infrastructure cost. Discussion on calculating infrastructure cost is beyond the scope of this paper. Every team is working on its specific component, which makes up the whole application. The evaluation and selection process of such elements is yet to be adopted for the blockchain community. Bali and Madan (2015a, 2015b), the authors of this paper, adopt a costing and performance evaluation method similar to that in the case of components of the shelf (COTS) for software development. Going by that COTS framework, the human resource cost can be calculated by assuming C1 to represent the cost incurred for blockchain architects, C2 as the cost for network administrators, C3 for channel administrators, C4 for ordering service administrators C5 for smart contracts developers. Equation (1) is the cost function considering all the variables and respective weightage.

We have studied the COCOMO intermediate model to derive or suggesting a better cost estimation for the effort put by our actors(C1, C2, C3, C4, C5), which indicate that there are additional cost factors that can drive the cost of the effort which gives by the manager by interviewing the actors.

COCOMO model categorises software in three kinds, namely, organic, semidetached and embedded. In this work, blockchain based lad registry system falls under the category of semidetached project. This is because the development team is composed of experienced as well as unexperienced members. As suggested in the study, it derives equation (2) where E is the effort, KLOC is kilo lines of code, effort adjustment factor (EAF), and b are the typical values for organic, semi-detached embedded software products, which has shown in Table 2. Initially, cost drivers have been identified and their values have been determined based on study by Boehm et al. (1995).

- Cost drivers: the value of EAF can be determined by looking into the cost drivers mentioned in the COCOMO design, shown in Table 3. Cost drivers are additional cost factors that accumulate to compute EAF as specified in equation (3). Mathematically, all the values of these cost drivers can be multiplied to get the EAF. Qualitatively, these cost drivers are concerned with workforce capability, the complexity of the system, and system constraints. The cost drivers have their impact, categorised as a range from very low to very high, with an equivalent range of numerical values as 0–2.0.

$$\text{Human Resource Cost} = 5C1 + 3C2 + 3C3 + 2C4 + C5 \tag{1}$$

$$\text{Intermediate Cost Estimation} = E = \left(a(KLOC)^b * EAF\right) \tag{2}$$

$$\text{EAF} = \Pi(\text{Cost Drivers}) \tag{3}$$

Table 2 Standard a and b values in COCOMO as proposed by Boehm et al. (1995)

Software projects	*a*	*b*
Organic	2.4	1.05
Semi-detached	3.0	1.12
Embedded	3.0	1.20

Table 3 Cost drivers and their impact on EAF computation

Cost drivers	*Very low (< 0.5)*	*Low (0.5 –1)*	*Normal (1)*	*High (1–1.5)*	*Very high (1.5–2.0)*
Required software reliability	0.23	0.67	1	1.12	1.56
The complexity of the constituent products	0.16	0.87	1	1.24	1.45
Memory constraints	-	-	1	1.06	1.21
Required turnaround time	-	0.94	1	1.23	1.34
Analyst capability	1.46	1.19	1	0.86	0.71
Software engineer capability	1.42	1.17	1	0.86	0.70
Use of software tools	1.24	1.10	1	0.91	0.82
Development Schedule	1.23	1.08	1	1.04	1.10

The function can be used to calculate those human resource costs that will incur for the respective roles. A keen observation that has to be emphasised is that the cost incurred for a blockchain architect is five times that of a smart contract developer. Hence, the total cost of developing a blockchain application is a factor of human resource cost and infrastructure, which is used for implementing the application. Such infrastructure can either be created on-premises, or can be utilised from service providers like Hyperledger, Amazon, IBM, and Oracle, to name a few. In the next section, the use case for this application is presented along with its implementation.

5 Use case: land registry system

According to the NITI Aayog's Draft Paper (2020) on Blockchain, the Land registry system in the country faces lots of issues. According to Mishra and Suhag (2017), there are several challenges to establishing clear land titles. Land titles are unclear because of being presumptive and lack of proper land record maintenance. A property needs to be registered for sale, transfer, gift, or lease following the Registration Act, 1908. As per Section 17 of the Registration Act, 1908, all transactions that involve the sale of immovable property for a value exceeding Rs 100 should be registered. If someone doesn't register a property while doing above mentioned transactions, then the transaction cannot be proved in a court of law. E-registration of property can be made simple by web-based applications which run using distributed ledger technology. This is a governance issue, which can be resolved using blockchain, as explained in the book by Morabito (2017). Thakur et al. (2020) have proposed a system design for implementing land titling in India, making land ownership clear and conclusive.

The proposed model is executed using dummy data. A case study has been constructed. There is a land registry authority in a district like Bangalore or Noida which works with multiple stakeholders like RERA, land registry office, titleHolder, government revenue office, land survey office. The registrar is responsible for providing unique identification (landID) to the land and relevant information like the first owner, currentOwner, size, location, value, date of transfer, etc. Every stakeholder works with each other in tandem and performs their transactions while keeping their transactions within their local ecosystem, agreeing to specific data sharing mechanisms or processes. Now the authors of this paper propose ZameenGyaan, a solution for these individual organisations to work together to be a part of a system where they can execute their transactions in a distributed or decentralised fashion which will help these organisations to:

1 Establish the ownership of land.

2 Manage the ownership transfer of land.

3 Track provenance of their transactions very quickly.

As per the proposed NBSOC framework and roles of the component teams, ZameenGyaan requires that the *blockchain architects design* the solution in a functional perspective, which further moves down to the *network administrators* to set up the network. They further configure the number of participants, identities, MSPs, certificate authority, IP addresses, and the port forwarding for the joining participants in the network. Once the network has been set up, *channel admins* configure the channels and induce the policies on the channels after discussing with all the participants and the architects. Channel will also hold the interface configurations for chain codes and ordering services. Further, *ordering service admins* will set up the ordering service nodes after understanding the volume data that will flow in the network. A huge performance lag can be encountered if the ordering service has not been configured according to the scenario. The transactions are bound to be in order before passing to the committing peers for committing them to the ledger. After all the configurations are done, *smart contract developers* are ready to write their business logic provided by the architect or the team leads after discussing with the network participants. Once smart contracts are deployed, installed, and instantiated to the network, peers, and channels, the blockchain network is ready to connect to the modular front-end of the decentralised applications.

6 Implementation of ZameenGyaan

The sale-purchase transaction (ownership transfer) is between the landowner and buyer is recorded on a ledger maintained in the blockchain. RERA and Land Survey offices can access the blockchain to verify and audit the land ownership and transfer authenticity. Figure 4 represents the application architecture of ZameenGyaan, where the Land Asset is created and stored by the Land Registry Office. When discussing about deployment of ZameenGyaan, it is vital to note further that third party blockchain service provider has been utilised for the purpose. There have been certain blockchain as a service platform which provide cloud storage, ledger storage, blockchain consensus and popular blockchain framework support. Xooa is the name of the aforementioned platform which provides support for Hyperledger fabric framework. In addition to this, it also supports chain code compilation and execution in the private blockchain network.

Figure 4 ZameenGyaan application architecture (see online version for colours)

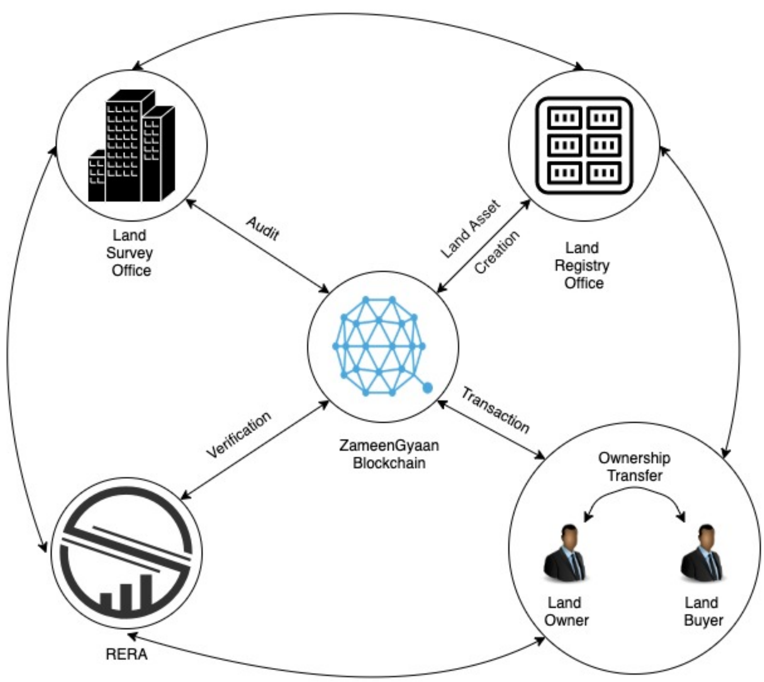


Figure 5 represents the implementation of ZameenGyaan, where ownership of land with ID RX1R5093 is established and stored in the blockchain. Other information related to the land is current owner, size, location, value, and DateofTransfer. The value of CurrentOwner changes whenever there is a change in the ownership of the land. This change in the value is again considered a transaction and stored in the immutable ledger of ZameenGyaan.

Figure 5 Creation of LandAsset (see online version for colours)

	Key	Value	Is Deleted	Updated At	Tx ID
▼	RX1R5093	docType	No	05/11/2020 1:16:12 am	1ac0f14b2a86c8e38166e660d68ef78dcf9fe d87d1c56133dd31082d6eadb1ff

docType	land
landId	RX1R5093
firstowner	Ramsharan
currentowner	Ramsharan
size	10 Marla
Location	11
Value	1 Cr
DateofTransfer	22/02/2019

Whenever a sale transaction has to be executed, the ownership transfer function mentions the landID, value, date of transfer, with the current owner now being Pradeep. Figure 6 represents the output of ownership transfer. The primary mechanism of this function is that it takes the landID, seller, and buyer as inputs. landID is the land asset that can be used to access all the previous transactions or changes made to the land. If the seller matches the current owner value mapped with landID, only then will ownership transfer happen. After execution and storing the information as transaction hashes in the new block, the new owner value is mapped to the buyer value and date and time of transfer. This generates TxID, which is the hash that maps to this transaction. Now, this information is broadcasted to all the peers in the blockchain.

Figure 6 Ownership transfer (see online version for colours)

	Key	Value	Is Deleted	Updated At	Tx ID
▼	RX1R5093	docType	No	05/26/2020 9:14:21 pm	6770f1aee7c9e00034b4c5ca6f99f9e79 f148755d17df1e1f750ed1f1e3790cb

docType	land
landId	RX1R5093
firstowner	Ramsharan
currentowner	pradeep
size	10 Marla
Location	11
Value	1 Cr
DateofTransfer	2020/05/26T15:44:22

Figure 7 shows the output when the txID is being queried. Value of CurrentOwner has changed to Pradeep. In addition to this, the first owner of the land is also presented in every query. This addresses the question about identifying the ownership of land.

Figure 7 Query current owner (see online version for colours)


```
{
    "txId": "8d6711057182d606fcf480998a8fb62b848c00d840a3299270ff436b96cb1927",
    "payload":     {
       "DateofTransfer": "2020/05/26T15:44:22",
       "Location": 11,
       "Value": "1 Cr",
       "currentowner": "pradeep",
       "docType": "land",
       "firstowner": "Ramsharan",
       "landId": "RX1R5093",
       "size": "10 Marla"
    }
}
```


7 Conclusions

From the presented discussion, this paper proposed NBSOC, a framework for the organisation of team structure, and concluded that there are two primary roles in developing any blockchain solution. One is the admin role on specific domains for the application of blockchain. The second is the developer role responsible for writing the codes for the decentralised applications and enabling them to interact with the blockchain's ledger using smart contracts.

In implementing the ZameenGyaan Solution, the two research questions regarding the team and infrastructural components required to implement blockchain applications for the specific use case have been addressed. For RQ1, a detailed team structure and individual roles and responsibilities have been identified and proposed in this paper, named the *NBSOC* framework. Network architects create the blockchain platform for the smart contract developers, managed by ordering admins and channel admins. Such a research question is necessary for blockchain software engineers to construct the required teams with the expected expertise. These domain experts can be indicative to determine the budgeting requirements before designing phase begins.

For RQ2, this paper presents an overview of the practical implementation of blockchain applications for human resources required and the flow of the application via the individuals working on it. ZameenGyaan has been used as an example implementation for the land registry use case in the Indian scenario. The blockchain architect is specific for this use case. Other roles are more or less the same in the horizontal scope of any domain. These roles are dependent on the individual's skills and knowledge about the different components of the application.

In addition to the proposed framework, the authors have also hinted at a blockchain cost pyramid, enabling managers to frame the appropriate costing model while developing a blockchain-based application. This is also represented using equations about how enterprises can utilise this pyramid and combine it with the COCOMO model to generate relevant costing information.

The future holds good promise for the technology stack of blockchain as a whole for providing a trust framework over a network of untrusting parties. Earlier, the transactions which involved participants trusting a third party for verification and validation can now count the blockchain system to validate the transactions by design. The current adoption of blockchain for the land titling use case cannot see the light of day until proper government regulations are declared. Clear guidelines need to be laid out before using Aadhaar and KYC for blockchain-based governance.

References

Adhi Tama, B., Kweka, B., Park, Y. and Rhee, K.H. (2017) 'A critical review of blockchain and its current applications', *International Conference on Electrical Engineering and Computer Science (ICECOS)*, at Palembang, Indonesia, pp.109–113, DOI: 10.1109/ ICECOS.2017.8167115.

Ahram, T., Sargolzaei, A., Sargolzaei, S., Daniels, J. and Amaba, B. (2017) 'Blockchain technology innovations', *IEEE Technology & Engineering Management Conference (TEMSCON)*, Santa Clara, CA, pp.137–141, DOI: 10.1109/TEMSCON.2017.7998367.

Anascavage, R. and Davis, N. (2018) 'Blockchain technology: a literature review', *SSRN Electronic Journal*, DOI: 10.2139/ssrn.3173406.

Andrian, H.R., Novianto, B.K. and Suhardi (2018) 'Blockchain technology and implementation: a systematic literature review', *International Conference on Information Technology Systems and Innovation (ICITSI)*, pp.370–374, DOI: 10.1109/ICITSI.2018.8695939.

Bali, V. and Madan, S. (2015a) 'COTS evaluation & selection process in design of component-based software system: an overview and future direction', *Global Journal of Computer Science and Technology*, Vol. 15, No. 6, pp.1–8, E-ISSN: 0975-4172, P-ISSN: 0975-4350 [online] https://computerresearch.org/index.php/computer/article/view/1280/1267 (accessed 11 April 2021).

Bali, V. and Madan, S. (2015b) 'Identifying and classifying factors for evaluating the success of software developed by COTS components', *International Journal of Current Engineering and Technology*, Vol. 5, No. 4, pp.2535–2541, E-ISSN: 2277-4106, P-ISSN: 2347-5161 [online] https://inpressco.com/wp-content/uploads/2015/07/Paper432535-2541.pdf (accessed 11 April 2021).

Bali, V., Khanna, T., Soni, P., Gupta, S., Chauhan, S. and Gupta, S. (2021) 'Blockchain application design and algorithms for traceability in pharmaceutical supply chain', *International Journal of Healthcare Information Systems and Informatics*, Vol. 16, No, 4, pp.1–18, DOI: 10.4018/ IJHISI.289460.

Baza, M., Nabil, M., Ismail, M., Mahmoud, M., Serpedin, E. and Rahman, M.A. (2019) 'Blockchain-based charging coordination mechanism for smart grid energy storage units', *IEEE International Conference on Blockchain (Blockchain)*, pp.504–509, https://arxiv.org/ abs/1811.02001.

Beller, M. and Hejderup, J. (2019) 'Blockchain-based software engineering', *IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, Montreal, QC, Canada, pp.53–56, DOI: 10.1109/ICSE-NIER.2019.00022.

Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. and Selby, R. (1995) 'Cost models for future software life cycle processes: COCOMO 2.0', *Annals of Software Engineering*, Vol. 1, pp.57–94, https://doi.org/10.1007/BF02249046.

Bosu, A., Iqbal, A., Shahriyar, R. and Chakrtaborty, P. (2019) 'Understanding the motivations, challenges, and needs of blockchain software developers: a survey', *Empirical Software Engineering*, Vol. 24, pp.2636–2673, DOI: 10.1007/s10664-019-09708-7.

Chakraborty, P., Shahriyar, R., Iqbal, A. and Bosu, A. (2018) 'Understanding the software development practices of blockchain projects: a survey', in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '18)*, Association for Computing Machinery, New York, NY, USA, Article 28, pp.1–10, DOI: 10.1145/3239235.3240298.

Crowston, K., Li, Q., Wei, K., Eseryel, U.Y. and Howison, J. (2007) 'Self-organization of teams for free/libre open source software development', *Information and Software Technology*, Vol. 49, No. 6, pp.564–575, ISSN: 0950-5849, DOI: 10.1016/j.infsof.2007.02.004.

Dabbagh, M., Sookhak, M. and Safa, N.S. (2019) 'The evolution of blockchain: a bibliometric study', in *IEEE Access*, Vol. 7, pp.19212–19221, DOI: 10.1109/ACCESS.2019.2895646.

Destefanis, G., Marchesi, M., Ortu, M., Tonelli, R., Bracciali, A. and Hierons, R. (2018) 'Smart contracts vulnerabilities: a call for blockchain software engineering?', *International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, Campobasso, pp.19–25, DOI: 10.1109/IWBOSE.2018.8327567.

Dinh, T.T.A., Wang, J., Chen, G., Liu, R., Ooi, B.C. and Tan, K. (2017) 'BLOCKBENCH: a framework for analyzing private blockchains', in *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD' 17)*, Association for Computing Machinery, New York, NY, USA, pp.1085–1100, DOI: 10.1145/3035918.3064033.

García-Bañuelos, L., Ponomarev, A., Dumas, M. and Weber, I. (2017) 'Optimized execution of business processes on blockchain', in Carmona, J., Engels, G. and Kumar, A. (Eds.): *Business Process Management. BPM 2017. Lecture Notes in Computer Science*, Vol. 10445, Springer, Cham, https://doi.org/10.1007/978-3-319-65000-5_8

Guo, H., Li, W., Nejad, M. and Shen, C. (2019) 'Access control for electronic health records with hybrid blockchain-edge architecture', *IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, pp.44–51, https://arxiv.org/abs/1906.01188.

Hajdu, Á., Ivaki, N., Kocsis, I., Klenik, A., Gönczy, L., Laranjeiro, N., Madeira, H. and Pataricza, A. (2020) 'Using fault injection to assess blockchain systems in the presence of faulty smart contracts', *IEEE Access*, Vol. 8, pp.190760–190783, DOI: 10.1109/ACCESS.2020.3032239.

Hegedus, P. (2018) 'Towards analyzing the complexity landscape of solidity based ethereum smart contracts', *IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Gothenburg, Sweden, pp.35–39 [online] https://ieeexplore.ieee.org/document/8445056 (accessed 18 December 2020).

Kaur, D. and Bali, V. (2015) 'Generating test cases from UML class and sequence diagrams', *International Journal of Research in Computer Science and Electronics*, Vol. 4, No. 1, pp.1–5, ISSN: 2319-376X.

Khanna, T., Nand, P., and Bali, V. (2020) 'Permissioned blockchain model for end-to-end trackability in supply chain management', *International Journal of e-Collaboration (IJeC)*, Vol. 16, No.1, pp.45–58, DOI: 10.4018/IJeC.2020010104.

Kizildag, M., Dogru, T., Zhang, T., Mody, M.A., Altin, M., Ozturk, A.B. and Ozdemir, O. (2019) 'Blockchain: a paradigm shift in business practices', *International Journal of Contemporary Hospitality Management*, Vol. 32, No. 3, pp.953–975, DOI: 10.1108/IJCHM-12-2018-0958.

Koul, R. (2018) 'Blockchain oriented software testing – challenges and approaches', *3rd International Conference for Convergence in Technology (I2CT)*, Pune, pp.1–6, DOI: 10.1109/I2CT.2018.8529728.

Kumar, A., Mahindru, T., Shukla, P., and Sharan, A. (2020) *Blockchain: The India Strategy*, NITI Aayog [online] https://niti.gov.in/sites/default/files/2020-01/Blockchain_The_India_Strategy_Part_I.pdf (accessed 18 January 2021).

Le-Dang, Q. and Le-Ngoc, T. (2019) 'Scalable blockchain-based architecture for massive IoT reconfiguration', *IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pp.1–4, DOI: 10.1109/CCECE.2019.8861858.

Liao, C., Cheng, C., Chen, K., Lai, C., Chiu, T. and Wu-Lee, C. (2017) 'Toward a service platform for developing smart contracts on blockchain in BDD and TDD styles', *IEEE 10th Conference on Service-Oriented Computing and Applications (SOCA)*, Kanazawa, 2017, pp.133–140, DOI: 10.1109/SOCA.2017.26.

Malik, S., Dedeoglu, V., Kanhere, S.S. and Jurdak, R. (2019) 'TrustChain: trust management in blockchain and IoT supported supply chains', *IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, pp.184–193, DOI: 10.1109/Blockchain.2019.00032.

Marchesi, M., Marchesi, L. and Tonelli, R. (2018) 'An agile software engineering method to design blockchain applications', in *Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia (CEE-SECR '18)*, Association for Computing Machinery, New York, NY, USA, Article 3, pp.1–8, DOI: 10.1145/3290621.3290627.

Mingxiao, D., Xiao-Feng, M., Zhe, Z., Xiang-Wei, W. and Qi-Jun, C. (2017) 'A review on consensus algorithm of blockchain', *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp.2567–2572, DOI: 10.1109/smc.2017.8123011.

Mishra, P. and Suhag, R. (2017) *Land records and titles in India*, PRS [online] https://www.prsindia.org/sites/default/files/parliament_or_policy_pdfs/Land%20Records%20and%20Titles%20in%20India.pdf (accessed 18 January 2021).

Mitani, T. and Otsuka, A. (2020) 'Traceability in permissioned blockchain', in *IEEE Access*, Vol. 8, pp.21573–21588, DOI: 10.1109/ACCESS.2020.2969454.

Morabito, V. (2017) *Business Innovation Through Blockchain: The B³ Perspective*, Springer [onlien] https://blockchainstudies.org/files/Morabito.pdf (accessed 10 December 2020).

Mylrea, M. and Gourisetti, S.N.G. (2017) 'Blockchain for smart grid resilience: exchanging distributed energy at speed, scale, and security', *Resilience Week (RWS)*, Wilmington, DE, pp.18–23, DOI: 10.1109/RWEEK.2017.8088642.

Nakamoto, S. (2008) *Bitcoin: A Peer-to-Peer Electronic Cash System* [online] https://bitcoin.org/bitcoin.pdf (accessed 17 October 2020).

Perboli, G., Musso, S. and Rosano, M. (2018) 'Blockchain in logistics and supply chain: a lean approach for designing real-world use cases', in *IEEE Access*, Vol. 6, pp.62018–62028, DOI: 10.1109/ACCESS.2018.2875782.

Pongnumkul, S., Siripanpornchana, C. and Thajchayapong, S. (2017) 'Performance analysis of private blockchain platforms in varying workloads', *26th International Conference on Computer Communication and Networks (ICCCN)*, Vancouver, BC, pp.1–6, DOI: 10.1109/ICCCN.2017.8038517.

Porru, S., Pinna, A., Marchesi, M. and Tonelli, R. (2017) 'Blockchain-oriented software engineering: challenges and new directions', *IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, Buenos Aires, pp.169–171, DOI: 10.1109/ICSE-C.2017.142.

Pradeep, D.S., Singi, K., Kaulgud, V. and Podder, S. (2018) 'Evaluating complexity and digitizability of regulations and contracts for a blockchain application design', *IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Gothenburg, Sweden, pp.25–29 [online] https://ieeexplore.ieee.org/document/8445054.

Puthal, D., Malik, N., Mohanty, S. Kougianos, E. and Das, G. (2018) 'Everything you wanted to know about the blockchain: its promise, components, processes, and problems', *IEEE Consumer Electronics Magazine*, Vol. 7, pp.6–14, DOI: 10.1109/MCE.2018.2816299.

Rocha, H. and Ducasse, S. (2018) 'Preliminary steps towards modeling blockchain oriented software', I*EEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Gothenburg, Sweden, pp.52–57 [online] https://ieeexplore.ieee.org/document/8445060 (accessed 20 May 2021).

Rouhani, S. and Deters, R. (2017) 'Performance analysis of ethereum transactions in a private blockchain', *8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, pp.70–74, DOI: 10.1109/ICSESS.2017.8342866.

Seebacher, S. and Schüritz, R. (2019) 'Blockchain – just another IT Implementation? A comparison of blockchain and interorganizational information systems', *European Conference on Information Systems (ECIS 2019)*, At Stockholm, Sweden.

Sillaber, C., Waltl, B., Treiblmaier, H., Gallersdörfer, U. (2020) 'Laying the foundation for smart contract development: an integrated engineering process model', *Information Systems and e-Business Management*, DOI: 10.1007/s10257-020-00465-5

Singh, S., Bhatnagar, V., Bali, V. and Bali, S. (2022) 'Optimization of intrusion detection systems determined by ameliorated HNADAM-SGD algorithm', *Electronics*, Vol. 11, No. 4, p.507, https://doi.org/10.3390/electronics11040507.

Somy, N.B., Kannan, K., Arya, V., Hans, S., Singh, A., Lohia, P., Mehta, S. (2019) 'Ownership preserving AI market places using blockchain', *IEEE International Conference on Blockchain (Blockchain)*, Atlanta, GA, USA, pp.156–165, DOI: 10.1109/Blockchain.2019.00029.

Sukhwani, H., Wang, N., Trivedi, K.S. and Rindos, A. (2018) 'Performance modeling of hyperledger fabric (permissioned blockchain network)', *IEEE 17th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, pp.1–8, DOI: 10.1109/NCA.2018.8548070.

Swan, M. (2015) 'Blockchain', *Blueprint for a New Economy*, O'Reilly, Sebastopol, CA.

Tanwar, S., Parekh, K. and Evans, R. (2020) 'Blockchain-based electronic healthcare record system for healthcare 4.0 applications', *Journal of Information Security and Applications*, Vol. 50, DOI: 10.1016/j.jisa.2019.102407.

Tasca, P. and Tessone, C. (2018) *Taxonomy of Blockchain Technologies*. *Principles of Identification and Classification* [online] SSRN: https://ssrn.com/abstract=2977811 or http://dx.doi.org/10.2139/ssrn.2977811 (accessed 21 April 2021).

Thakur, V., Doja, M.N., Dwivedi, Y.K., Ahmad, T., and Khadanga, G. (2020) 'Land records on blockchain for implementation of land titling in India', *International Journal of Information Management*, Vol. 52, DOI: 10.1016/j.ijinfomgt.2019.04.013.

Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E. and Alexandrov, Y. (2018) 'SmartCheck: static analysis of ethereum smart contracts', *IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Gothenburg, Sweden, pp.9–16 [online] https://ieeexplore.ieee.org/document/8445052 (accessed 25 November 2020).

Tsiulin, S., Reinau, K.H., Hilmola, O-P., Goryaev, N. and Karam, A. (2020) 'Blockchain-based applications in shipping and port management: a literature review towards defining key conceptual frameworks' *Review of International Business and Strategy*, Vol. 30, No. 2, pp.201–224, DOI: 10.1108/RIBS-04-2019-0051.

Vingerhouts, A.S., Heng, S. and Wautelet, Y. (2020) 'Organizational modeling for blockchain oriented software engineering with extended-i* and UML', *CEUR Workshop Proceedings*, Vol. 2749, pp.23–34 [online] http://ceur-ws.org/ (accessed 25 January 2021).

Viswanathan, R., Dasgupta, D., and Raghavan, S.G. (2019) 'Blockchain solution reference architecture (BSRA)', *IBM Journal of Research and Development*, p.1, DOI: 10.1147/jrd.2019.2913629.

Vora, J., Nayyar, A., Tanwar, S., Taygi, S., Kumar, N., Obaidat, M.S. and Rodrigues, J.J.P.C. (2018) 'BHEEM: a blockchain-based framework for securing electronic health records', *IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, pp.1–6, DOI: 10.1109/GLOCOMW.2018.8644088.

Wang, G., Shi, Z., Nixon, M., and Han, S. (2019) 'Chain splitter: towards blockchain-based industrial IoT architecture for supporting hierarchical storage', *IEEE International Conference on Blockchain (Blockchain)*, pp.166–175 [online] https://arxiv.org/abs/1910.00742 (accessed 20 February 2021).

Wessling, F. and Gruhn, V. (2018) 'Engineering software architectures of blockchain-oriented applications', *IEEE International Conference on Software Architecture Companion (ICSA-C)*, Seattle, WA, pp.45–46, DOI: 10.1109/ICSA-C.2018.00019.

Wessling, F., Ehmke, C., Hesenius, M. and Gruhn, V. (2018) 'How many blockchains do you need? Towards a concept for building hybrid DApp architectures', *2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, Gothenburg, Sweden, pp.44–47 [online] https://ieeexplore.ieee.org/abstract/document/8445058 (accessed 25 February 2021).

Wüst, K. and Gervais, A. (2018) 'Do you need a blockchain?', *Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp.45–54, DOI: 10.1109/CVCBT.2018.00011.

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Puatasso, C. and Rimba, P. (2017) 'A taxonomy of blockchain-based systems for architecture design', *IEEE International Conference on Software Architecture (ICSA)*, Gothenburg, pp.243–252, DOI: 10.1109/ ICSA.2017.33.

Yli-Huumo, J., Ko, D., Choi, S., Park, S. and Smolander, K. (2016) 'Where is current research on blockchain technology? – a systematic review', *PLoS ONE*, Vol. 11, No. 10, p.e0163477, DOI: 10.1371/journal.pone.0163477.

Yu, S., Lv, K., Shao, Z., Guo, Y., Zou, J., and Zhang, B. (2018) 'A high-performance blockchain platform for intelligent devices', *1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Shenzhen, pp.260–261, DOI: 10.1109/ HOTICN.2018.8606017.

Zheng, Z., Xie, S., Dai, H., Chen, X. and Wang, H. (2017) 'An overview of blockchain technology: architecture, consensus, and future trends', *IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, pp.557–564, DOI: 10.1109/BigDataCongress.2017.85.

Zheng, Z., Xie, S., Dai, H-N., Chen, X. and Wang, H. (2018) 'Blockchain challenges and opportunities: a survey', *International Journal of Web and Grid Services*, Vol. 14, No. 4, pp.352–375, https://doi.org/10.1504/IJWGS.2018.095647.