IJNTI
International Journal of Novel Trends and Innovation
IJNTI.ORG | ISSN : 2984-908X
An International Open Access, Peer-reviewed, Refereed Journal

SMART TENDER/CONTRACT MANAGEMENT SYSTEM USING BLOCKCHAIN

[1] MIRWAISE KHAN, [2] TAMILARASAN.M, [3] AJMAL BAIG ABDULLA, [4] NAREN JANSON

[1] Student, [2] Student, [3] Student, [4] Student

[1] Department of Computer Science & Engineering,

[1] Rajiv Gandhi Institute of Technology, Bangalore, India

Abstract : Generally, the Tenders or contracts are used by governments and companies to procure goods or services. Wrongful tender management leads to huge losses in case of faulty practices. This includes favouring of contractors, improper record maintenance, lack of transparency, hacking, data modification and other issues. To overcome this problem, we have used a simple and secure block chain technology and to secure by encryption coupled with indisputable block based architecture for transaction management. In this case we make use of block chain technology to secure transaction based documents along with transactions such as tender documents, applications, bid proposals, company profiles, past records, approving officer details, rejection details to ensure a completely transparent tendering process.

KEYWORDS: Block chain, Tenders, Bidders,Contractors.

1. INTRODUCTION

Current E-Tendering systems are not ‘fair and open’ meaning that the information is not shared with all stakeholders. The information is released on ‘as they please’ basis for example - when a company is selected as a winner of a contract, other companies that bid on the same tender are not notified of why their bid was rejected and why a particular company was selected as a winner. A company can request this information but it is a tedious process of getting this data. Even though auditing these documents is possible, evaluating the documents needs time. Apart from not being transparent, security is also a major issue for these portals leading to fraud and manipulation of data stored in a centralized database. If a hacker gets hold of this centralized database, bids can be leaked to competitors leading to major financial and strategic losses for a business.

Blockchain technology can be used to solve these security implications as it heavily focuses on the decentralization of information and is secured by encryption integrated with undeniable blockbased architecture for transaction management. Hence, Blockchain and Smart Contract can be used as a transparent, decentralized and secured tendering framework that will facilitate bidders' oversight on portal functions and observe all the activities carried out by the tender portal.

Blockchain Explained Blockchain is based on the concept of decentralization. Hence, it can be viewed as a distributed database. In this case, the distributed database employs the concept of full replication i.e. each node has a full copy of a blockchain. Whenever the blockchain needs to be updated because of a transaction, a process called mining takes place . A block consists of many transactions. A consensus protocol is used and the mined block is broadcasted to all other nodes. These blocks will have a cryptographic hash in the header that relates to the previous block in the chain. If a block is manipulated the hash associated with this block changes and as a result, all the proceeding blocks should be re-mined which is not possible. In this manner, blockchain employs the property of immutability. How the blockchain is implemented and what consensus protocol is the core of blockchain.

2. NEED OF THE STUDY.

7.1 Feasibility Study The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- ♦ Economical feasibility
- ♦ Technical feasibility
- ♦ Social feasibility

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 Types of Tests

7.2.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2.3 Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items: Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised. Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2.4 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.2.5 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated,

as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

3. Related Work:

[1] Wang, Wenbo, et al. "A survey on consensus mechanisms and mining strategy management in blockchain networks." IEEE Access 7 (2019): 22328-22370.

The past decade has witnessed the rapid evolution in blockchain technologies, which has attracted tremendous interests from both the research communities and industries. The blockchain network was originated from the Internet financial sector as a decentralized, immutable ledger system for transactional data ordering. Nowadays, it is envisioned as a powerful backbone/framework for decentralized data processing and data-driven self-organization in flat, open-access networks. In particular, the plausible characteristics of decentralization, immutability, and self-organization are primarily owing to the unique decentralized consensus mechanisms introduced by blockchain networks. This survey is motivated by the lack of a comprehensive literature review on the development of decentralized consensus mechanisms in blockchain networks. In this paper, we provide a systematic vision of the organization of blockchain networks. By emphasizing the unique characteristics of decentralized consensus in blockchain networks, our in-depth review of the stateof-the-art consensus protocols is focused on both the perspective of distributed consensus system design and the perspective of incentive mechanism design. From a game-theoretic point of view, we also provide a thorough review of the strategy adopted for self-organization by the individual nodes in the blockchain backbone networks. Consequently, we provide a comprehensive survey of the emerging applications of blockchain networks in a broad area of telecommunication. We highlight our special interest in how the consensus mechanisms impact these applications. Finally, we discuss several open issues in the protocol design for blockchain consensus and the related potential research directions.

Summary: Wang, Wenbo and team describes in this paper, we provide a systematic vision of the organization of blockchain networks

[2] Ambegaonker, Ajeenkkya, Utkarsh Gautam, and Radha Krishna Rambola. "Efficient approach for Tendering by introducing Blockchain to maintain Security and Reliability."
2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.

The problem with present tendering is its reach which is limited to number of people, though the internet is expanding and tendering is also not far from this, we have some online system for tendering but it is not secure as it should be because tendering has confidential data which is not supposed to be leaked and Blockchain solves that problem efficiently. The motive of this research is to find the better ways for tendering, as tendering is very essential part of businesses and development so improvement of this system leads to better development. Time efficiency, employment, fair system are some of the factors which can be improved by the proposed system of this research.

Summary: Ambegaonker, Ajeenkkya and team working on online system for tendering but it is not secure as it should be because tendering has confidential data which is not supposed to be leaked and Blockchain solves that problem efficiently.

[3] Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE international congress on big data (BigData congress). IEEE, 2017.

Blockchain, the foundation of Bitcoin, has received extensive attentions recently. Blockchain serves as an immutable ledger which allows transactions take place in a decentralized manner. Blockchain-based applications are springing up, covering numerous fields including financial services, reputation system and Internet of Things (IoT), and so on. However, there are still many challenges of blockchain technology such as scalability and security problems waiting to be overcome. This paper presents a comprehensive overview on blockchain technology. We provide an overview of blockchain architechture firstly and compare some typical consensus algorithms used in different blockchains. Furthermore, technical challenges and recent advances are briefly listed. We also lay out possible future trends for blockchain.

Summary: Zibin and team provide an overview of blockchain architechture firstly and compare some typical consensus algorithms used in different blockchains.

[4] Cachin, Christian, and Marko Vukolić. "Blockchain consensus protocols in the wild." arXiv preprint arXiv:1707.01873 (2017).

A blockchain is a distributed ledger for recording transactions, maintained by many nodes without central authority through a distributed cryptographic protocol. All nodes validate the information to be appended to the blockchain, and a consensus protocol ensures that the nodes agree on a unique order in which entries are appended. Consensus protocols for tolerating Byzantine faults have received renewed attention because they also address blockchain systems. This work discusses the process of assessing and gaining confidence in the resilience of a consensus protocols exposed to faults and adversarial nodes. We advocate to follow the established practice in cryptography and computer security, relying on public reviews, detailed models, and formal proofs; the designers of several practical systems appear to be unaware of this. Moreover, we review the consensus protocols in some prominent permissioned blockchain platforms with respect to their fault models and resilience against attacks.

Summary: Christian, and team discusses the process of assessing and gaining confidence in the resilience of a consensus protocols exposed to faults and adversarial nodes.

[5] Pilkington, Marc. "Blockchain technology: principles and applications." Research handbook on digital transformations. Edward Elgar Publishing, 2016.

This paper expounds the main principles behind blockchain technology and some of its cutting-edge applications. Firstly, we present the core concepts at the heart of the blockchain, and we discuss the potential risks and drawbacks of public distributed ledgers, and the shift toward hybrid solutions. Secondly, we expose the main features of decentralized public ledger platforms. Thirdly, we show why the blockchain is a disruptive and foundational technology, and fourthly, we sketch out a list of important applications, bearing in mind the most recent evolutions.

Summary: Pilkington and team expose the main features of decentralized public ledger platforms.

[6] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 254–269.

Cryptocurrencies record transactions in a decentralized data structure called a blockchain. Two of the most popular cryptocurrencies, Bitcoin and Ethereum, support the feature to encode rules or scripts for processing transactions. This feature has evolved to give practical shape to the ideas of smart contracts, or full-fledged programs that are run on blockchains. Recently, Ethereum's smart contract system has seen steady adoption, supporting tens of thousands of contracts, holding millions dollars worth of virtual coins. In this paper, we investigate the security of running smart contracts based on Ethereum in an open distributed network like those of cryptocurrencies. We introduce several new security problems in which an adversary can manipulate smart contract execution to gain profit. These bugs suggest subtle gaps in the understanding of the distributed semantics of the underlying platform. As a refinement, we propose ways to enhance the operational semantics of Ethereum to make contracts less vulnerable. For developers writing contracts for the existing Ethereum system, we build a symbolic execution tool called Oyente to find potential security bugs. Among 19, 366 existing Ethereum contracts, Oyente flags 8, 833 of them as vulnerable, including the TheDAO bug which led to a 60 million US dollar loss in June 2016. We also discuss the severity of other attacks for several case studies which have source code available and confirm the attacks (which target only our accounts) in the main Ethereum network.

Summary: L. Luu, D.-H and team introduce several new security problems in which an adversary can manipulate smart contract execution to gain profit

4. SYSTEM ANALYSIS & FEASIBILITY STUDY

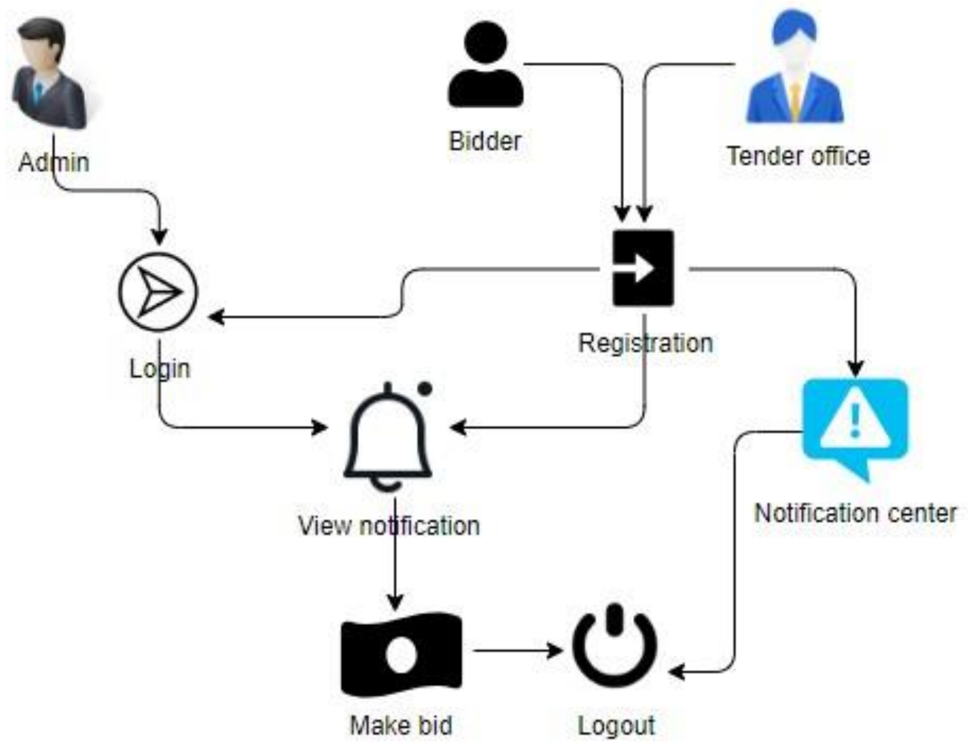
EXISTING METHOD:

In the existing system, all the work is done manually. Contractors need to submit their documents on time and must be submitted through ordinary post by which they sometimes not able to bid for particular tender on time. All working personnel within department involved just for doing the same task which is document verification and there may be a chance in which the best one may be left behind.

PROPOSED SYSTEM:

The Proposed Tender Management System uses block chain technology to ensure the complete tender management process is secure and efficient. A block chain is secured by encryption coupled with indisputable block based architecture for transaction management. This allows the system to maintain a simple transparent transaction with need-to-know basis information conveying.

work Flow of Proposed system:



In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships

The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data? The data to be encrypted. This array we call the state array. You take the following aes steps of encryption for a 128-bit block: Derive the set of round keys from the cipher key. Initialize the state array with the block data (plaintext). Add the initial round key to the starting state array.

Perform nine rounds of state manipulation. Perform the tenth and final round of state manipulation. Copy the final state array out as the encrypted data (ciphertext). The reason that the rounds have been listed as "nine followed by a final tenth round" is because the tenth round involves a slightly different manipulation from the others. The block to be encrypted is just a sequence of 128 bits. AES works with byte quantities so we first convert the 128 bits into 16 bytes. We say "convert," but, in reality, it is almost certainly stored this way already. Operations in RSN/AES are performed on a two-dimensional byte array of four rows and four columns. At the start of the encryption, the 16 bytes of data.

5. REQUIREMENT ANALYSIS:

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS:

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and nonfunctional requirements.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements. They basically deal with issues like: • Portability • Security • Maintainability • Reliability • Scalability • Performance • Reusability • Flexibility Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
2) The processing of each request should be done within 10 seconds
3) The site should load in 3 seconds whenever of simultaneous users are > 10000

SYSTEM SPECIFICATIONS:

Hardware System Configuration:

- • Processor - I3/Intel Processor
• RAM - 4GB (min)
• Hard Disk - 160GB

Software System Configuration:-

• Operating System : Windows 7/8/10
• Application Server : Xampp
• Front End : HTML, CSS
• Scripts : JavaScript.
• Database : My SQL
• Technology : Python 3.9+.

6. SYSTEM DESIGN:

5.1 Introduction of Input design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc. Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties –

• It should serve specific purpose effectively such as storing, recording, and retrieving the information.
• It ensures proper completion with accuracy.
• It should be easy to fill and straightforward.
• It should focus on user's attention, consistency, and simplicity.

- **All these objectives are obtained using the knowledge of basic design principles regarding –**

o What are the inputs needed for the system?
o How end users respond to different elements of forms and screens.

OBJECTIVES FOR INPUT DESIGN: The objectives of input design are –

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

OUTPUT DESIGN: The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

OBJECTIVES OF OUTPUT DESIGN: The objectives of input design are: • To develop output design that serves the intended purpose and eliminates the production of unwanted output. • To develop the output design that meets the end user's requirements. • To deliver the appropriate quantity of output. • To form the output in appropriate format and direct it to the right person. • To make the output available on time for making good decisions.

UML DIAGRAMS

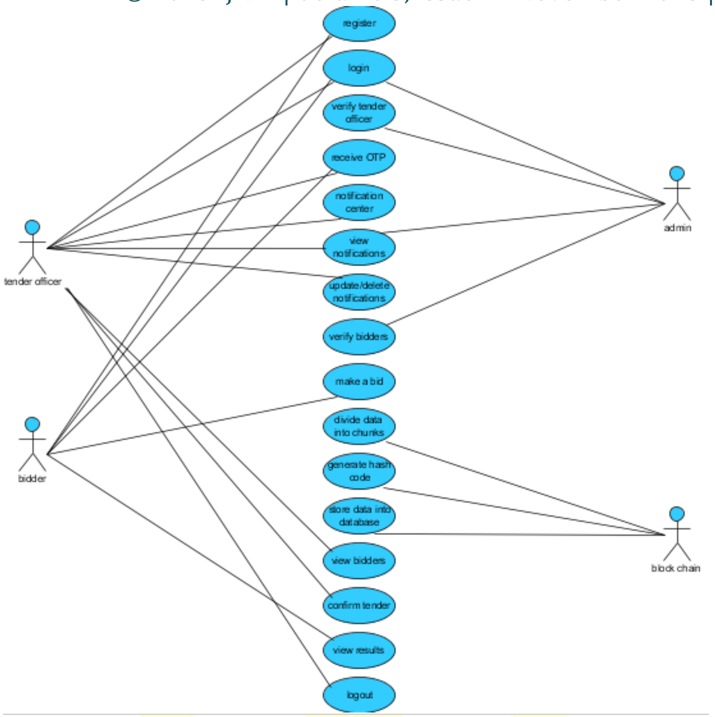
UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:
1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models. 2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM

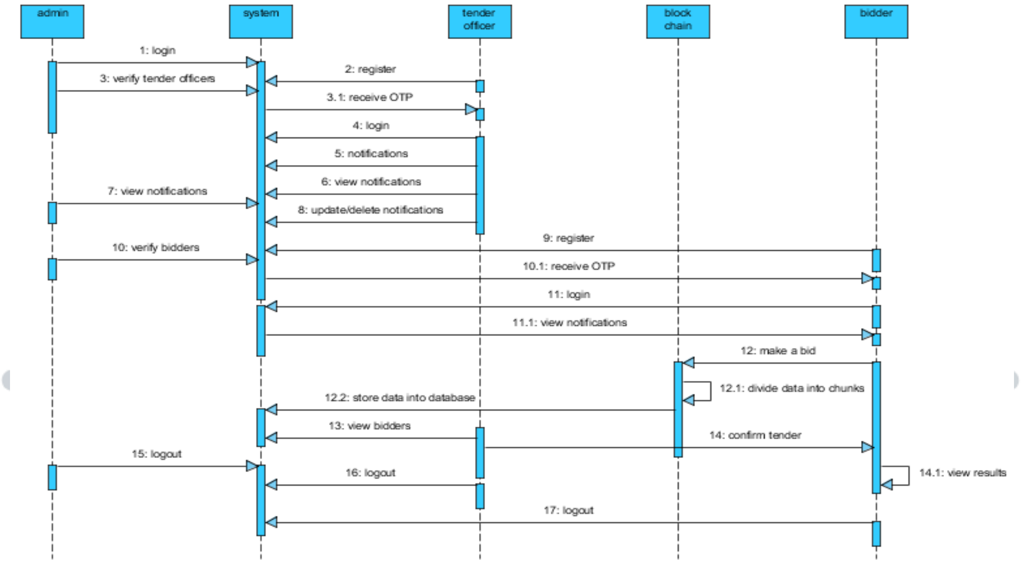
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



SEQUENCE DIAGRAM

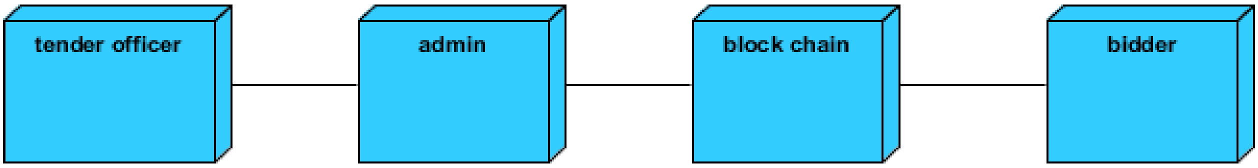
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.

It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

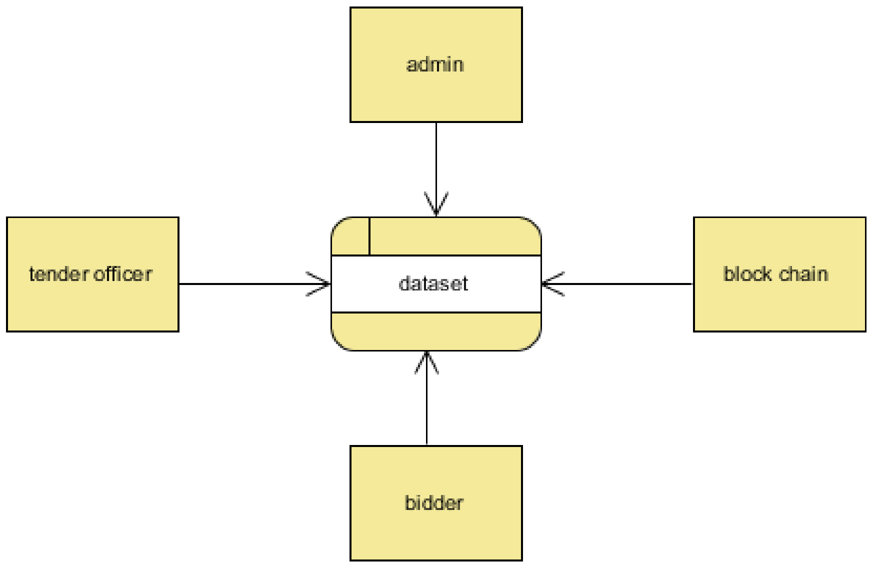


DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.



Context Level Diagram:



7. IMPLEMENTATION AND RESULTS

MODULES:

The proposed system includes 3 entities: tender officers, bidders and blockchain. Figure 1 shows the interaction between each entity in the proposed system

❖ **Tender Officer:** Tender officer will login into the account after registration and update the notification regarding the tender process. There is a option where they can modify or delete the notification part. Now the officer will download the tender files for which were register by the bidders and decrypt the data from downloaded files to get the information of bidders. Once after getting the information of bidders a confirmation mail sent to the bidders as the acceptance for their tender applications.

❖ **Bidder:** Bidders will login into the account after registration and they will view the tender notifications. If the bidder is okay with tender description he/she will provide their information in text file to the bidder officer. After sending the application they can check the response from the tender officer.

❖ can make a tender to the tender officer by providing data in a text file.

❖ **Blockchain:** The blockchain is used to store an encrypted formate by dividing the data into chunks. Here apply the hash code on chunks for hiding the data after converting it into an encrypted format which is stored in a database.

8. CONCLUSION:

When it comes to applications such as tender portals, where transparency and security are of foremost importance, traditional technologies and design patterns cannot be used as they put a threat to these requirements. As discussed earlier, there are many security requirements for a tendering framework that cannot be solved just by using a centralized tender portal for creating and bidding on the contracts. The security requirements and openness required from this type of application can only be solved by using fair, open, decentralized technology such as Blockchain and Smart Contracts. In this paper, how such a system can be designed by mentioning various processes involved and their basic implementation.

9. FUTURE SCOPE:

There are two further research directions, which are as follows - The Smart Contract can be made more secure by using more complex cryptographic algorithms for eg. SHA-256 to encrypt its confidential contents. The use of blockchain is explored further in other government services.

10. REFERENCES:

1. K. C. Davis, “The information act: A preliminary analysis,” TheUniversity of Chicago Law Review, vol. 34, no. 4, pp. 761– 816, 1967.

2. Ambegaonker, Ajeenkkya, Utkarsh Gautam, and Radha Krishna Rambola. "Efficient approach for Tendering by introducing Blockchain to maintain Security and Reliability." 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.

3. Pal, Om, and Surendra Singh. "Blockchain Technology and Its Applications in E-Governance Services."

4. Betts, Martin, et al. "Towards secure and legal e-tendering." Journal of Information Technology in Construction 11 (2006): 89- 102

5. Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE international congress on big data (BigData congress). IEEE, 2017.

6. Pilkington, Marc. "Blockchain technology: principles and applications." Research handbook on digital transformations. Edward Elgar Publishing, 2016.

7. Wang, Wenbo, et al. "A survey on consensus mechanisms and mining strategy management in blockchain networks." IEEE Access 7 (2019): 22328-22370.

8. Cachin, Christian, and Marko Vukolić. "Blockchain consensus protocols in the wild." arXiv preprint arXiv:1707.01873 (2017).

9. Cuccuru, Pierluigi. "Beyond bitcoin: an early overview on smart contracts." International Journal

10. L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 254–269.

11. K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi,G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote,N. Swamy et al., “Formal verification of smart contracts: Short paper,”in Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security. ACM, 2016, pp. 91–96

12. Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.