Proceedings of the ASME 2021
International Mechanical Engineering Congress and Exposition
IMECE2021
November 1-5, 2021, Virtual, Online

IMECE2021-71730

EVALUATING THE IMPLEMENTATION OF DISTRIBUTED LEDGER TECHNOLOGY FOR THE LICENSING AND REGULATION OF NUCLEAR POWER PLANTS

Priyanka Pandit
North Carolina State University, Raleigh, NC

Alp Tezbasaran
North Carolina State University, Raleigh, NC

Arjun Earthperson
North Carolina State University, Raleigh, NC

Mihai A. Diaconeasa
North Carolina State University, Raleigh, NC

ABSTRACT

The approval process from the U.S. Nuclear Regulatory Commission (NRC) for nuclear power plants is sequential. It involves several government bodies such as the Advisory Committee on Reactor Safeguards (ACRS), public meetings, and hearings. If the submissions made to the NRC do not contain enough information to meet the regulation requirements, the NRC issues a Request for Additional Information (RAI). Thus, the licensee has to go through a paperwork-intensive process that involves multiple regulatory agencies for the various licensing requirements. Moreover, sending applications to the NRC is limited to using an electronic submission generation tool called the Packing Slip Wizard (PSW).

This paper presents a methodology to implement Distributed Ledger Technology (DLT) to address the need for a real-time, digitized documentation platform in the nuclear power industry's licensing and regulation process.

The evaluation of DLT's implementation resulted in the formulation of a methodology to accept submissions from an applicant on a web application and storing the received data on a distributed ledger. The presented method offers a real-time submission of the available information of an application. It facilitates the NRC with a real-time feedback capability expediting the review process. RAI's can be reduced in number by ensuring that the NRC's information requirements are defined as smart contracts.

Keywords: Blockchain, Nuclear Power Plants, Hyperledger, Licensing, and Regulation.

NOMENCLATURE

ACRS	Advisory Committee on Reactor Safeguards
ADAMS	Agencywide Documents Access and Management System
API	Application Programming Interface
COLA	Combined Operating and Construction License
DLT	Distributed Ledger Technology
MSP	Membership Service Provider
NRC	Nuclear Regulatory Commission
PSW	Packing Slip Wizard
RAI	Request for Additional Information
SDK	Software Development Kit
XML	eXtensible Markup Language

1. INTRODUCTION

The U.S. Nuclear Regulatory Commission (NRC) is the regulatory body that oversees the licensing and regulation of nuclear power plants. The NRC issues licenses such as construction permits, operating licenses, early site permits, standard design certification, combined license, manufacturing license, duplicate plant license, standard design approval, and site suitability reviews. It is responsible for establishing rules, licensing, oversight, enforcement, evaluation, incident response,

and support provision concerning civilian nuclear projects [1]. 10 CFR Part 50 is the two-step licensing process that approves a construction permit and an operating license [2]. 10 CFR Part 52 deals with other licensing processes such as early site permits, standard design certification, combined license, manufacturing license, duplicate plant license, standard design approval, and site suitability reviews [3]. 10 CFR Part 53 is under development, encapsulating the new frameworks required for licensing and regulating advanced nuclear reactors [4].

The approval process is sequential and involves several government bodies such as the Advisory Committee on Reactor Safeguards (ACRS), public meetings, and hearings [5]. For the issuance of a combined license, the licensee must meet specific Inspections, Tests, Analyses, and Acceptance Criteria standards called ITAAC [6]. If the submissions made to the NRC do not contain enough information to meet the regulation requirements, the NRC issues a Request for Additional Information (RAI) [7]. Thus, the licensee must go through a paperwork-intensive process that involves multiple regulatory agencies for the various licensing requirements.

Moreover, sending applications to the NRC is limited to using an electronic submission generation tool called the Packing Slip Wizard (PSW). The wizard generates an eXtensible Markup Language (XML) file for submission to the NRC. Dispatching physical documentation copies to the NRC is a time-consuming process. The use of e-mail communication facilitates the reduction in the transport of documents; however, it still retains the need to generate the electronic submission file using the packing wizard. As a result, volumes of physical paper trails have to be developed, classified, archived, and tracked by the licensee.

For this paper, A DLT based web application was designed and demonstrated to accept submissions from the user and maintain them on a distributed ledger. The application's front-end collected data and documentation using an interface similar to the packing slip wizard provided by the NRC. The Hyperledger fabric network formed the backend. Its function was to maintain the activity of the network's members in the form of transactions.

The submission fields required to represent a topic area completely were defined as the terms of a smart contract. The submission status on the web application changes to complete only if all the necessary information is provided. The NRC has oversight over all the transactions on the network. In comparison, the licensees' access is limited to their correspondence with the NRC.

The process currently employed by the NRC to accept electronic submissions utilizes the PSW to prepare an application at the licensee end [8]. The operating system requirements used to run the wizard are Windows XP or Vista, which are very limiting nowadays. Once the software is installed from the NRC website, the submission can be prepared. The wizard has various submission fields that store data or documents pertaining to that field, as shown in FIGURE 1.

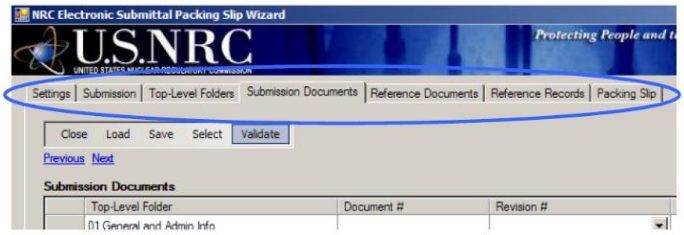

FIGURE 1: TOPIC AREAS IN THE PACKING SLIP WIZARD

The submission tab on the wizard specifies the applicant's identity generated, as shown in FIGURE 2.

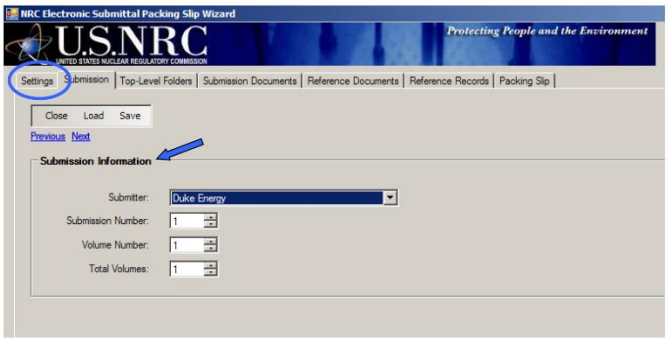

FIGURE 2: SUBMISSION TAB OF THE PACKING SLIP WIZARD

Every individual section of the application is a separate Top-Level Folder. For example, the top-level folders for a combined operating and construction license (COLA) are shown in FIGURE 3.

d:\01 General and Admin Info
d:\02 FSAR
d:\03 Environmental Report
d:\04 Tech Specs
d:\05 Emergency Plan
d:\06 LWA-Site Redress Plan
d:\07 DCD Departures
d:\08 Safeguards and Security
d:\09 Plant Specific PRA
d:\10 ITAAC

FIGURE 3: TOP-LEVEL FOLDERS FOR COLA

The documents contained in the application are uploaded in the *Submission Documents* Tab. References used in the submissions are specified in the *Reference Document* and *Reference Records* tabs. After correctly supplying all the information in the previous tabs, the packing slip tab generates an XML file, as shown in FIGURE 4. Then the XML file is copied and burned onto a CD/DVD. If revisions are to be made to the submission, then a new XML file needs to be generated containing the revised documents and an indication of the updated fields.

FIGURE 4: XML FILE GENERATED BY THE PACKING SLIP WIZARD

All the submission documents received by the NRC are stored on NRC's official record-keeping system Agencywide Documents Access and Management System (ADAMS). The files have a unique identifier for archival and version control [9].

2. INTRODUCTION TO BLOCKCHAIN

This section introduces the reader to the critical concepts needed for the implementation of the blockchain technology.

2.1 Definition of Blockchain

A blockchain can be defined as an immutable ledger for recording data transactions maintained within a distributed network of nodes or peers. Each peer maintains its copy of the entire ledger and continuously executes what is known as a consensus protocol to validate past and new transactions, group them into a new block to be added to a node, and then hash the new block to enforce immutability and consistency. The hash binds all new blocks successively to their preceding blocks such that every node has only one preceding parent node - which ensures consistency across all the peer nodes in the blockchain system[4]. Blockchain is a subset of Distributed Ledger Technology, referred to as DLT from this point onwards[5]. DLT's main selling point is the ability to create and establish immutable digital record-keeping systems without relying on a centralized database, thus resulting in transparency and assurance of consistency for all entities and parties involved in the system.

2.2 Public and Private Blockchains

Blockchains are of two kinds: permissionless and permissioned systems [6]. Permission-less or public blockchains are blockchains that require no permissions or authorizations to join and interact with. Hence, they are commonly used for managing cryptocurrencies [7]. The first and most famous public blockchain is Bitcoin, enabling users to transfer digital currencies between themselves without needing a single third-party financial institution or centralized database to validate transactions. Such systems are decentralized in the true sense of the word, with users free to interact with one another with a great degree of anonymity and transparency. On the other hand, permissioned blockchain systems are private, meaning not everyone can join the network without authorization [8]. These are closed networks cut off from the public. Permissioned systems are used when an organization or a consortium of organizations cannot afford to make their processes and data publicly available while maintaining a great degree of inter-organizational trust, transparency, and immutability of transactions. Permissioned systems are used by banks and other such organizations requiring a greater degree of data security while having a varying degree of decentralization. In contrast to public blockchains, permissioned systems can choose their consensus methods without requiring all the nodes in the network for validation, making transaction speeds faster.

2.3 Advantages of Blockchain

DLT, or more specifically blockchain, has the potential to strengthen data integrity and security. As organizations generate, store, and exchange an increasing amount of data of varying sensitivity levels, the importance of protecting this organizational data from adversarial attacks, sabotage, or even from internal human error also grows. For this reason, Blockchain technologies and applications have piqued the interest of stakeholders in a wide variety of fields [4].

2.4 Immutability of Blockchain

One of blockchain's most important properties is the immutability of the data and transaction blocks and is fundamentally defined within blockchain's basic architecture. This is ensured through multiple components such as hashing of successively linked data blocks, digital signatures and time stamps implemented on the ledger, proof-of-work algorithms for authorizing entities on the network to add new blocks, consensus protocols, and mechanisms, and most importantly, the distributed network of untrusted peer nodes. Each of these components acts as a series of failsafe against potential data block inconsistency, tampering, adversarial attacks, unauthorized third-party entities adding data blocks on the network, and so on [9].

2.5 Hyperledger Fabric

Building on several previously outlined principles, Hyperledger Fabric is an open-source platform for distributed ledgers with a highly modular architecture designed to be private and permissioned [10]. Rather than an open permission-less system like those used in conventional cryptocurrency exchanges, which require proof-of-work to validate transactions, Hyperledger Fabric uses a trusted Membership Service Provider (MSP) to validate trusted members on the network. Due to its modular nature, ledger data can be stored in multiple formats. Different consensus mechanisms can be swapped in and out of the network, and even support multiple MSPs simultaneously. Hyperledger Fabric supports the creation of multiple channels, where a group of participants can create a separate ledger of transactions. This is important for networks where some participants may not be authorized to view the transactions being made by other entities or would not want every transaction they made known to all participants.

The ledger system of Fabric consists of the world state and the transaction log. All participants in the Hyperledger have a copy of the entire ledger network to which they belong. The World State describes the state of the ledger at a given point in time, while the transaction log records all the transactions that have resulted in the current value of the world state. The ledger then combines the world state database and the transaction log history of the network. One could use Hyperledger Fabric to build a system for managing nuclear reactor construction between multiple construction stakeholders in a consortium while also allowing participants to manage how much oversight other stakeholders have over their additions to the blockchain network.

The direct use of the Blockchain ledger would be in creating a historical database of previously completed projects that the nuclear power plant construction stakeholders could use as an aid in Project Estimation tasks. This historical ledger can also be defined in Hyperledger Fabric, creating a contiguous and effective implementation of a Blockchain enabled Smart contract-based system for software development.

2.6 Smart Contracts

Before a project can start or before businesses and other entities begin working together and trade information, all parties involved must agree to adhere to a common set of contracts that cover terms, rules, process, and concept definitions. This collective set of contracts form the business model that all parties will follow, and is usually converted into actionable, executable programs called smart contracts. By implementing the smart contract, it allows users to automatically enforce the business model's governance rules on any business object or work product when the smart contract is executed. In the context of blockchain, smart contracts can be used to define and record data transactions to a Blockchain Ledger [11]. The smart contract architecture can help reduce the overall project risk of a nuclear reactor designer and owner by having an open communication channel with the regulators during the reactor design process to be able to iteratively obtain early feedback.

Hyperledger Fabric currently can handle multiple deep smart contracts written in standard, general-purpose programming languages, further making it easier to create smart contracts within a rapid time frame.

2.7 Shallow and Deep Smart Contracts

There are various degrees of automation involved in defining smart contracts, which in turn determine the level of implementation of blockchain with smart contract objects. These are shallow smart contracts and deep smart contracts. Shallow smart contracts have simplistic definitions for basic operations such as cryptocurrency exchanges between authorized entities. On the other hand, deep smart contracts are capable of handling multiple complex contracts and rule sets that are dependent on the nature or the input data object or trigger. These deep smart contracts are used to define the transaction logic that controls the lifecycle of a business object or work product. The deep smart contract defines the object and its transition through different states and records the changes in states using *chaincode* that is then deployed on the Blockchain network [11].

3. PROPOSED METHODOLOGY

In recent years, Distributed Ledger Technologies (DLT) have found practical uses beyond conventional cryptocurrencies in the healthcare, education, and data security domains by enabling trusted environments for information sharing. The main advantage of distributed ledger technologies is their cryptographically secure properties. Unlike a single centralized database, the users can copy, update, and append operations to the data stored in all other connected nodes to the blockchain, verifying the added data's integrity. By continuously verifying copies of data with other peers on the blockchain, this process ensures that the information is secure and tamperproof. As a result, DLT offers advantages such as inter-organizational trust and secure digitization of paperwork. This paper's ledger system is an extensible, distributed, modular, access-controlled, and open-source ledger system implemented over the Hyperledger Fabric. Hyperledger Fabric differentiates itself from existing blockchain solutions in its approach to existing smart contracts instead of those defined with cryptocurrency or domain-specific languages [10].

To utilize Hyperledger Fabric, a web-based application process needs to be developed. Section 2.1 of the paper describes the architecture of the proposed Hyperledger Fabric network forming the back end. Section 2.2 describes the front end architecture of the proposed web application for nuclear regulatory engagement.

3.1 Hyperledger Fabric Network

The architecture of the Hyperledger Fabric network is defined by the components of the network, namely the ledger, peers, smart contract, channels, certificate authority, events, orderers, and systems management [11].

The ledger is the part of the network that maintains a log of the activity on the network, as shown in FIGURE 5. The activities being recorded on the network are called transactions, as depicted in FIGURE 6. Peers are the entities that make up the network and host ledgers and smart contracts [12]. The smart contract and the ledger form the engine that drives the Hyperledger Fabric. A smart contract is the business logic that needs to be satisfied for a transaction to be executed. The representation of the smart contract conditions in the form of executable code is called *chaincode* [13].

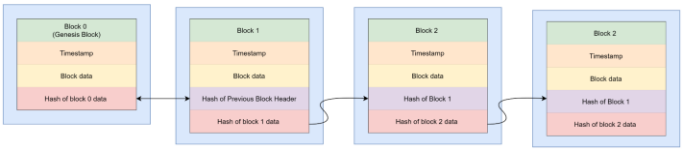

FIGURE 5: CHAIN OF BLOCK HEADERS THAT FORMS THE LEDGER

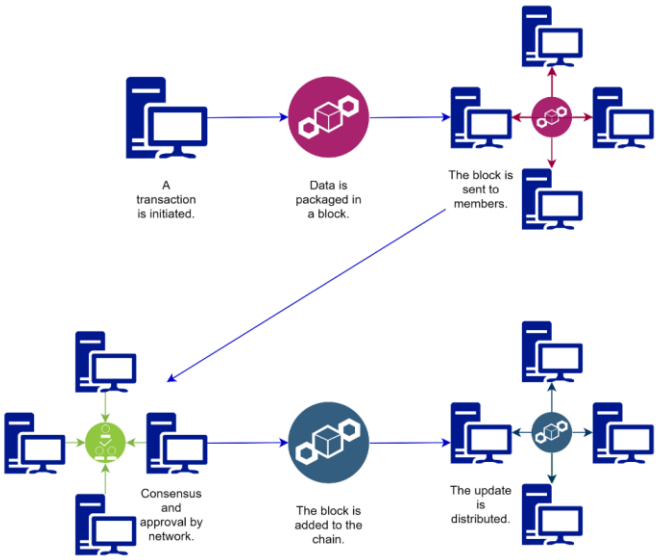

FIGURE 6: LEDGER BLOCK UPDATE PROCESS

Channels are created to keep transactions confidential within a set of peers that are members of a channel. The certificate authority is the network entity that provides identification keys and permissions to the peers on the network. When transactions events occur, the front end of the application emits a notification of the activity. Systems management is the management of the network and its entities; it can be either outsourced or done in-house. Orderers are entities on the network that can order the ledger to record a transaction.

Consider the example of a fabric network shown in FIGURE 7. The system manager begins with the deployment of the fabric network for the NRC and defines users within the NRC and ACRS as peers on the network. Additionally, the licensees are also specified as peers on the network. A single licensee organization can have multiple users registered on the network as peers.

A specific peer on the network is designated as the certificate authority. The certificate authority provides a unique identifier for each peer in the form of a private key and defines permissions for each peer. A private key can be thought of as a digital signature. Thus, the identities of all the participants on the network and their jurisdiction of data are known [14].

As shown in the example network in FIGURE 7, P1, P2, P3, …, P13 are peers belonging to the organizations, the NRC, the ACRS, and two licensees A and B. C1 is the channel that facilitates the interaction between the regulatory bodies and rest of the organizations on the network. Thus, the NRC can view the information stored by all the peers on the network. However, the ACRS peers and the licensee peers can only view the information they have been given access to by the certificate authority in the NRC. The various peers within an organization can view information internally using channels C2, C3, C4, and C5.

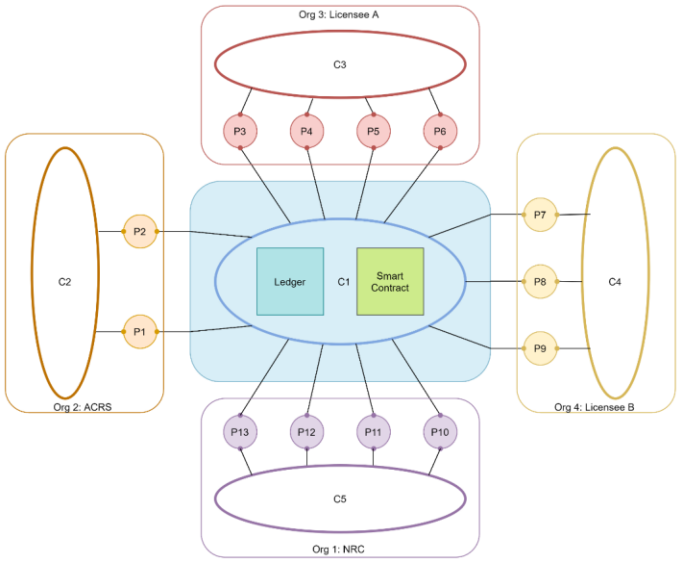

FIGURE 7: HYPERLEDGER-FABRIC NETWORK ARCHITECTURE EXAMPLE

The interaction process of the Hyperledger Fabric with the front end is facilitated by the fabric software development kit (SDK), containing an application programming interface (API) that will communicate back and forth between the front end of the application and the fabric network as shown in FIGURE 8. For our use case, we can define the terms of the smart contract to be the completion of all the information fields required by the NRC to review the application. For this purpose, the front end will have to be built, keeping the granularity of the information in mind. The NRC's information fields will have to be broken down to most fundamental questions that are then answered by the licensee on the web application. For the example case, a smart contract can intake the information equivalent to that presented in one chapter in one document in the packing slip wizard. Once the user on the licensee end inputs the application information into the web application, the fabric API invokes the *chaincode* with the input data. If all the submission fields are complete with the appropriate data type, then the *chaincode* accepts the API's proposal. The API then communicates with the orderer to record the transaction on the ledger. This process, as shown in FIGURE 8 is repeated with every peer on the network. Once the peers reach consensus via a pluggable mechanism, the transaction gets appended on each copy of the ledger. In FIGURE 8, P1 is the peer, S1 is the chaincode, L1 is the ledger, O1 is the orderer, and A is the front end.

3.2 Front End Architecture

To facilitate familiarity for the user, the architecture of the packing slip wizard can be replicated. The mock-up of a front end from a licensee point of view is shown in FIGURE 9.

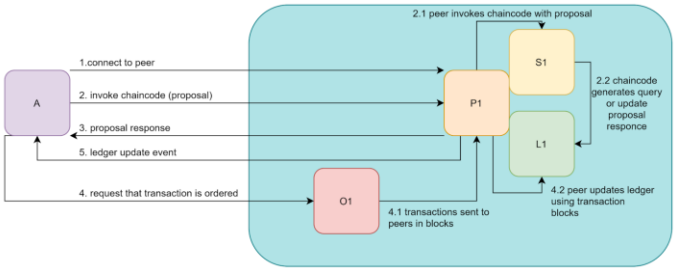

FIGURE 8: APPLICATION INTERACTION WITH THE NETWORK

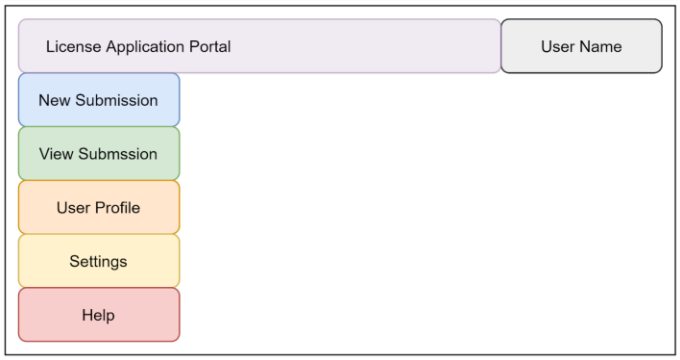

FIGURE 9: MOCK-UP OF FRONT END - LICENSEE LOGIN

The new submission field can have a drop menu, as shown in FIGURE 10. Say, for example, we select the tab COLA, then the front end should display all the topic areas associated with a COLA as shown in FIGURE 11. FIGURE 12 and FIGURE 13 show the view of the front end when the user navigates to FSAR>Design of structure.

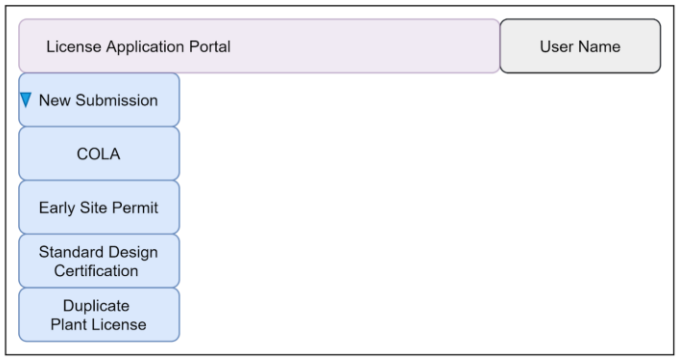

FIGURE 10: MOCK-UP OF FRONT-END SHOWING TYPES OF SUBMISSIONS

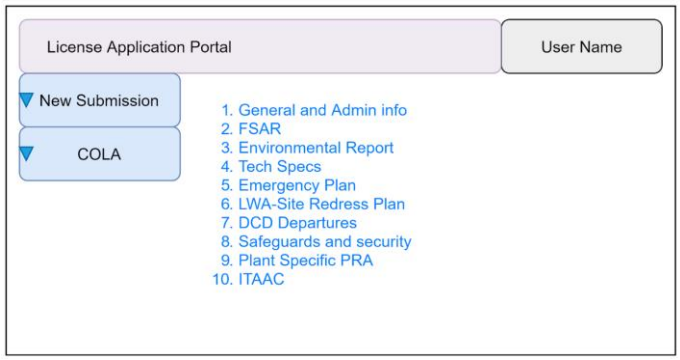

FIGURE 11: MOCK-UP OF SUBMISSION FIELDS FOR COLA

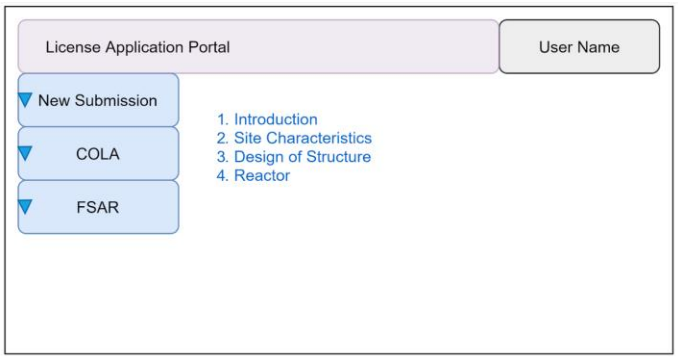

FIGURE 12: MOCK-UP OF SUBMISSIONS FIELD FOR FSAR

The front end can integrate NRC feedback for the submission fields at a granular level. If the NRC user submits feedback asking for changes in the submission, the licensee user can see the feedback in real-time and edit the submission field. Both the feedback and the modification of submission are transaction events and emit a notification for the respective users with a timestamp of the action as shown in FIGURE 13. These transactions also get recorded on the ledger to achieve version control for the application information, as shown in FIGURE 14.

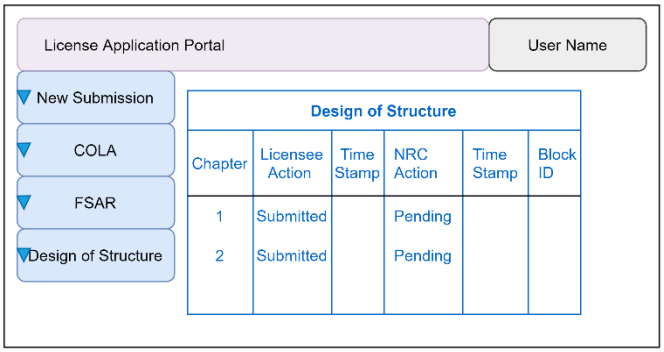

FIGURE 13: MOCK-UP OF SUBMISSION FIELDS FOR DESIGN OF STRUCTURE

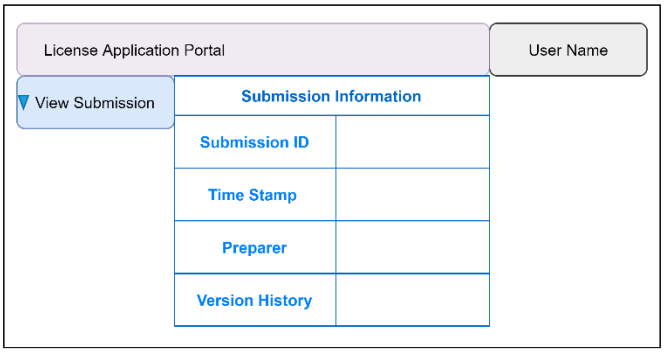

FIGURE 14: MOCK-UP OF VERSION CONTROL FOR SUBMISSION

4. DISCUSSION

Having presented a methodology to digitize the licensing process for nuclear power plants, we can now compare it to the current method for submitting applications.

The proposed DLT implementation reduces interaction friction in the licensing process by using a web-based application instead of physical submission media. The licensee's time and cost associated with the passage of the submission media to the NRC can be eliminated. A user-friendly interface can ease the process of applying for a license. The number of queries received by the point of contact or the project managers at NRC can be reduced due to the availability of the NRC feedback field on the web application. To share information with the ACRS, the NRC must only specify viewing permissions for ACRS users; again, the regulators need not use physical submission media. The licensee need not maintain a paper trail of documents for record-keeping, as the distributed ledger provides immutability and version control.

The proposed DLT implementation reduces innovation friction by facilitating the real-time submission of information about an application and when it is available at the licensee end. This enables the NRC to give their review on the progress of an application right from the beginning. The licensee can register various users within its organization onto the network such that these users are responsible for different parts of the licensing process. Measuring project success becomes easier for the upper management due to the visibility of the submission timestamps of users within the licensee organization. Accountability increases within all the organizations involved due to shared access to user identity submitting.

The ledger's cryptographic properties ensure that the submissions are made in a secure environment to protect intellectual property and user data.

5. CONCLUSION

This paper presented a methodology to implement DLT for licensing and regulation of nuclear power plants. The evaluation of the proposed methodology compared to the current method elicited many benefits in favor of a DLT platform. One of the most promising applications of the proposed methodology is to quantify project and licensing risk. The real-time visibility of the NRC's decisions from the beginning of the process will reduce the risk associated with the licensing of nuclear power plants.

ACKNOWLEDGEMENTS

The authors would like to thank the Department of Nuclear Engineering at North Carolina State University for fostering the spirit of innovation.

REFERENCES

[1] "NRC: Who we are and what we do." Accessed: Apr. 28, 2021. [Online]. Available: https://www.nrc.gov/docs/ML2000/ML20003E672.pdf

[2] "NRC: 10 CFR Part 50—Domestic Licensing of Production and Utilization Facilities." https://www.nrc.gov/reading-rm/doc-collections/cfr/part050/ (accessed Sep. 04, 2020).

[3] "US NRC: 10 CFR Part 52- Licenses, Certifications, and Approvals for Nuclear Power Pants." https://www.nrc.gov/reading-rm/doc-collections/cfr/part052/ (accessed Sep. 04, 2020).

[4] C. Matthew W. Sunseri, "'10 CFR Part 53 Licensing and Regulation of Advanced Nuclear Reactors' Available online: https://www.nrc.gov/docs/ML2029/ML20295A647.pdf," p. 6.

[5] "Nuclear Power Plant Licensing Process." Accessed: Apr. 28, 2021. [Online]. Available: https://www.nrc.gov/docs/ML0421/ML042120007.pdf

[6] "NRC Inspection Manual." Accessed: Apr. 28, 2021. [Online]. Available: https://www.nrc.gov/docs/ML1211/ML12110A239.pdf

[7] "Request for Addistional Information Job Aid." Accessed: Apr. 28, 2021. [Online]. Available: https://www.nrc.gov/docs/ML1819/ML18199A235.pdf

[8] "Document Intake and Review Process Packing Slip Wizard User's Guide," U.S. Nuclear Regulatory Commission, Oct. 2007. [Online]. Available: https://www.nrc.gov/site-help/e-submittals/packing-slip-wizard-user-guide-v1-1.pdf

[9] B. Golchane, "US NRC:Web-Based ADAMS User Guide. Available online: https://www.nrc.gov/reading-rm/adams/wba-user-guide.pdf," p. 44.

[10] E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *Proc. Thirteen. EuroSys Conf. - EuroSys 18*, pp. 1–15, 2018, doi: 10.1145/3190508.3190538.

[11] "Blockchain network — hyperledger-fabricdocs master documentation." https://hyperledger-fabric.readthedocs.io/en/release-2.2/network/network.html#the-sample-network (accessed Apr. 28, 2021).

[12] “Peers — hyperledger-fabricdocs master documentation.” https://hyperledger-fabric.readthedocs.io/en/release-2.2/peers/peers.html (accessed Apr. 28, 2021).
[13] “Key Concepts — hyperledger-fabricdocs master documentation.” https://hyperledger-fabric.readthedocs.io/en/release-2.2/key_concepts.html (accessed Apr. 29, 2021).
[14] “Identity — hyperledger-fabricdocs master documentation.” https://hyperledger-fabric.readthedocs.io/en/release-2.2/identity/identity.html (accessed Apr. 28, 2021).