Department of Computer Science and Engineering

University of Moratuwa

DYNAMIC SPECTRUM ACCESS VIA SMART CONTRACTS ON BLOCKCHAIN

Final Year Project Report

Group 04

B.P.T.D. Ariyarathna	140042R
P.N. Harankahadeniya	140205V
S. I. Isthikar	140238A
C.M.N.D. Pathirana	140436E

Supervisors

Dr. H.M.N. Dilum Bandara	University of Moratuwa
Associate Prof. Arjuna Madanayake	Florida International University

Abstract

Spectrum sharing using Dynamic Spectrum Access is becoming essential due to the ever-growing demand for the radio spectrum. Although there is a massive amount of bandwidth available at mm-waves, physics dictate the use of legacy frequencies in the sub 6 GHz range, which necessitates Dynamic Spectrum Access in the face of exponentially growing spectral demands. Disorganized spectrum sharing causes interference, leading to a chaotic situation and loss of capacity. Furthermore, it is not guaranteed that the primary users are compensated for sharing their licensed bands. A blockchain-based platform is proposed for enhancing the efficient use of the spectrum while addressing above limitations. A digital token, called *spectral token,* is introduced to enforce sequential access to spectrum by secondary users while avoiding interference, and to validate and track the use of a license of a particular frequency band. The proposed method supports both advertising and sensing based approaches for the initiation procedure for sharing spectrum. The licensed users have the privilege to customize the leasing policy coded into the smart contract based on either competitive bidding or first-come-first-served basis. Smart contracts digitally enforce the contractual clauses of the leasing agreement. A successful smart contract automatically transfers the *spectral token* between primary and secondary users within the agreed time frame while paying the primary user in native cryptocurrency. Blockchain acts as a platform which facilitates fast spectral token-based transactions between the primary and the secondary users in a secure, distributed way thereby avoiding the need for a trusted third party. This enhances the performance of the system by avoiding the involvement of intermediary parties in the transaction. We demonstrate the utility of the proposed solution by developing a proof of concept solution using Ethereum platform. Performance results show that the system has good throughput and latency characteristics.

Keywords—Blockchain; Cognitive Radio; Radio spectrum; Smart Contracts; Spectrum sharing

Acknowledgement

We would like to thank Prof. Arjuna Madanayake from Florida International University, USA firstly for his brilliant idea of the project and valuable insights on the subject matters throughout the project.

This project would have been impossible without the great support and guidance from our supervisor of the project, Dr. Dilum Bandara. His valuable advices and extraordinary assistance helped us greatly to make this project a success. So, many thanks sir for your great support and guidance.

Next we would like to thank Dr. Charith Chithraranjan, final year project coordinator and Dr. Shehan Perara Head of the Department of Computer Science and Engineering, University of Moratuwa for the support and guidance given throughout the project process.

We also would like to thank all the lecturers of the department for their advices and feedbacks on project in various situations like progress reviews and evaluations which gave us great help to improve the project.

Finally, we would like to thank all our colleagues of the department and all the other people who helped us even with a word, throughout the project duration to make this project a success.

Table of Contents

List of Figures

List of Abbreviations

Abbreviation	Description
CBRS	Citizens Broadband Radio Service
CLI	Command Line Interface
CPU	Central Processing Unit
CR	Cognitive Radio
DSA	Dynamic Spectrum Access
FCC	Federal Communications Commission
FCFS	First Come First Served
IEEE	Institute of Electrical and Electronic Engineers
ITO	Initial Token Offering
ISM band	Industrial, Scientific, and Medical radio band
JSON	JavaScript Object Notation
LTE	Long Term Evolution
LTE-LAA	LTE Licensed Assisted Access
LTE-U	LTE Unlicensed
MAC	Medium Access Control
PAL	Priority Access License
PoA	Proof of Authority
PoC	Proof of Concept

PoS	Proof of Stake
PoW	Proof of Work
QoS	Quality of Service
RAM	Random Access Memory
RPC	Remote Procedure Call
Spass	Spectrum Sensing as a Service
TRC	Telecommunications Regulatory Commission
TTBV	Two-Threshold based Voting
USA	United States of America
WRAN	Wireless Regional Area Networks

1 INTRODUCTION

The rapid evolution communication technologies have led to tremendous growth in wireless system capacity and global penetration. Consequently, the number of mobile services and users increase significantly creating exponentially rising demands for radio spectrum [1]. For example, Long Term Evolution (LTE) is expanding its utilization in unlicensed band by deploying LTE Unlicensed (LTE-U) and Licensed Assisted Access LTE (LTE-LAA) technologies to conquer high demand for faster data rates [2]. Moreover, limitations imposed by affordable communication infrastructure and establishment of next-generation 5G communication standard [3], has made the increasing commercial auctioning of mm-wave bands, making radio frequency spectrum a vastly valuable, scarce, natural resource. Most of the current wireless networks are characterized by a static spectrum allocation policy, where regulatory agencies, such as the Federal Communications Commission (FCC), in USA, and the Telecommunications Regulatory Commission (TRC) in Sri Lanka, assign radio spectrum to license holders on a long-term basis for large geographical regions. However, these license holders use the radio spectrum sporadically, sometimes hardly ever using them, causing unused *spectrum holes* (aka., *white spaces*) [4]. These spectrum holes can appear and disappear in a range of time scales, starting from a few milliseconds and going up to several weeks. However, the radio spectrum possesses an outrageous price due to the speculated demand-supply gap, as the radio spectrum is already heavily congested [5] due to the licensing mode that ignores spectrum holes.

Cognitive radio [6] aims to solve the problem by dynamically assigning the spectrum holes to secondary users as they become available. However, primary users who paid large sums of money for guaranteed use of the bands, are reluctant to allow secondary users to exploit their spectrum due to technical and economic/risk management reasons. Also, the interference that may occur due to secondary transmitting in a frequency band which belongs to a licensed user leads the licensed user to take legal actions against the secondary user. Therefore, the transactions that involve spectrum sharing should be done

by maintaining integrity and transparency between licenses users, secondary users and the regulatory body overseeing legal use of the spectrum.

It is mandatory to ensure that the spectrum sharing mechanisms are well organized, because it permits many parties to transmit information via a frequency channel, causing interference which leads to a chaotic situation if not coordinated correctly between all players. The traditional solution to avoid interference is the to make use of spectrum sensing to avoid collisions with the primary user. Interference occurs when primary user initiates the transaction while the secondary user is transmitting [6] because the primary user is not aware of the current user of the spectrum and he is not also bothered to check it because he has the right to transmit information anytime as he has the legal ownership for it. Moreover, cognitive radio in its basic form does not guarantee that the primary users are paid for the respective frequency bands exploited by the secondary users. This is because there is no restriction for cognitive radios to use white spaces and they do not bother to detect the owners of the respective frequency bands used. Also, the owners of the frequency band used by the cognitive radio would not detect that it is being used by cognitive radio unless they start transmitting in the same frequency band and interference occurs.

In this work, we propose digital token based dynamic spectrum sharing platform using blockchain and Smart Contracts technologies to improve the spectrum utilization and the security of the radio spectrum access. Blockchain is a decentralized technology that ensures the integrity of the exchanged data while eliminating any single-point of failure [7]. The ledger which stores the data is accessible, verifiable and auditable. These features of the blockchain intercepts the involvement of a trusted third-party during transactions. All the transactions happening via blockchain are witnessed by every node in the blockchain network, hence supporting the transparency and authenticity. A consensus protocol on blockchain such as proof of work (PoW) or proof of stake (PoS) together with cryptography can add security and fraud protection to the transactions. Also, some of the blockchain platforms such as Ethereum [8] and Hyperledger [9] embeds a feature called

smart contracts which helps to incorporate the necessary conditions that should be considered while completing the transactions. The smart contracts in Ethereum facilitates to use it as an account and indisputably enforces contractual clauses. These features motivated us to use Ethereum instead of Hyperledger.

1.1 Motivation

Radio spectrum is a scarce natural resource with increasing demand due to exponentially increasing usage for mobile and broadband services. As mentioned above the license to transmit data in the spectrum is issued by government regulatory bodies typically for one time or for a long time period. Currently most of the spectrum is already allocated and sold out. The licensing cost per frequency band is very high due to its demand. But primary users do not use spectrum always leaving vacant frequency bands which are called white spaces causing underutilization of spectrum. Cognitive radio is a viable solution for this problem. Cognitive radios can intelligently detect white spaces and transmit data in them. But there are some limitations in cognitive radios such as no guaranteed payments and uncertainty of misuse causing interference.

Cognitive Radio (CR) [6] aims to solve the demand-supply gap by dynamically assigning the spectrum holes to secondary users as they become available. For example, Citizens Broadband Radio Service (CBRS) [10] enables incumbent, primary user, and secondary user access based on three priority tiers. Primary users could purchase frequency bands from FCC via competitive bidding, while primary users may lease unused frequency bands to secondary users enabling a secondary market. Both primary users and secondary users must honor the incumbent's use where frequency bands need to be released as and when needed by the incumbent. While such Dynamic Spectrum Access (DSA) enables better utilization of the radio spectrum while reducing the overall cost, it is nontrivial to coordinate the multi-party, policy-driven, and timely spectrum sharing. Disorganized DSA could lead to a chaotic situation increasing interference while reducing the Quality of Service (QoS) and capacity. For example, standards such as IEEE 802.22 Wireless

Regional Area Networks (WRAN) mandate that secondary communication with CR in TV broadcast band [11] does not lead to harmful interference to the incumbent's operation. Moreover, primary users who paid large sums of money for the guaranteed use of the frequency bands, are reluctant to allow secondary users to exploit their spectrum due to the lack of trust on secondary users, uncertainty in payments, and complexity and cost of formal contracts which are unappealing given the short time span of DSA. Therefore, the transactions that involve DSA should protect the access to the frequency bands and maintain integrity, transparency, and trust between the primary users, secondary users, and regulatory body overseeing the legal use of the spectrum.

The idea of combining CR with spectrum sensing [12] and Blockchain [7] technology could potentially solve many of the issues that prevent the practical use of CR while facilitating standards such as the IEEE 802.22 WRAN and CBRS. For example, the applicability of blockchain technology for DSA in CBRS is discussed in [10]. CBRS is a 150 MHz wide broadcast band (from 3550 MHz to 3700 MHz) in USA which grants a leasing process as permitted by the FCC, to enable secondary market's spectrum usage rights held by licensed CBRS users [12]. The CBRS sharing framework consists of three tiers of spectrum access, and Priority Access License (PAL) is issued for three years through a competitive bidding process. These Priority Access licensees are free to lease their license to secondary users to maximize the spectrum utilization. However, the author does not present a potential implementation. In [13], authors present a blockchain-based verification protocol to gain better throughput over ALOHA MAC protocol while reducing the need for continuous spectrum sensing. Moreover, the author proposed to pay the primary users for sharing their spectrum via virtual currency. However, this solution also lacks an implementation and supports only primary user initiated and first-come-first-served (FCFS) style bidding. Spass [12] is a blockchain-based solution to incentivize wide-spread spectrum sensing where spectrum sensors are rewarded in virtual currency for accurate identification of spectrum holes. While physical spectrum sensing is essential to detect violations of DSA by secondary users, the number of spectrum sensors and sensing resolution could be reduced by keeping track of spectrum allocations on the

blockchain. Use of blockchain-based smart contracts to negotiate between mobile operators and domestic users to host Small-Cell as a Service node is presented in [14]. While each of the related work addresses complementary aspects of blockchain-based DSA, there is a still a need to develop and test a unified platform that could support different spectrum sharing and bidding policies. Therefore, we propose a blockchain-based DSA platform to address above issues.

1.2 Problem Statement

Problem to be addressed by this research can be formulated as follows:

> *Can dynamic spectrum sharing be done using blockchain and smart contract technologies to utilize radio spectrum ensuring permissioned access?*

1.3 Objectives

Above problem statement is to be addressed by achieving following objectives:

1. To devise a solution to make the transactions between primary & secondary users fast and guaranteed without the involvement of a third-party using blockchain & smart contract technologies.
2. To develop a bidding platform for the negotiation between primary & secondary spectrum users.
3. To support primary-driven spectrum advertising, secondary-driven spectrum identification, & varying bidding & pricing rules.
4. To develop a proof of concept solution & demonstrate its utility & performance.

1.4 Outline

The rest of the report is organized as follows. Chapter 2 presents the literature review. Chapter 3 discusses the methodology of the system under the subtopics system overview and system design. The implementation of the system including the front end, blockchain and smart contracts is discussed in chapter 4. The Chapter 5 of the report describes the Empirical evaluation where the Proof of Concept (PoC) and results obtained are analytically discussed. The Conclusion which is in the Chapter 6 reflects an overview of the system, limitations and the future works of the proposed system.

2 LITERATURE REVIEW

2.1 Spectrum Management

Various traditional methods of spectrum management have been discussed in [16]. As in this paper, auctions have been used as a mechanism of selling goods and services for thousands of years. The earliest example of auctions can be traced back to AD 195 when Roman Empire was auctioned off to Julianus. There are numerous examples of Auctions in our daily life, but auctions caught the imagination of economists, governments and common people alike when spectrum rights were auctioned by FCC in 1994 in the USA. Prior to using auctions for spectrum allocation other methods like administrative process, lottery and first-come-first serve were widely used. Since FCC's auction a number of countries switched to auctions for spectrum allocation because of their comparative advantages over other methods. This paper critically analysis the pros and cons of different methods used in spectrum allocation. The method of using Simultaneous Ascending Auctions in allocation of spectrum is discussed along with the advantages that may accrue from this method.

The paper [17] mentions that cognitive radio networks will provide high bandwidth to mobile users via heterogeneous wireless architectures and dynamic spectrum access techniques. However, cognitive radio networks impose challenges due to the fluctuating

nature of the available spectrum, as well as the diverse QoS requirements of various applications. It is emphasized that the spectrum management functions can address these challenges for the realization of this new network paradigm. To provide a better understanding of cognitive radio networks, this article presents recent developments and open research issues in spectrum management in cognitive radio networks. More specifically, the discussion is focused on the development of cognitive radio networks that require no modification of existing networks. First, a brief overview of CR and the CR network architecture is provided. The four main challenges of spectrum management are discussed: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility.

CR is the enabling technology for supporting dynamic spectrum access: the policy that addresses the spectrum scarcity problem that is encountered in many countries as mentioned in [18]. Thus, CR is widely regarded as one of the most promising technologies for future wireless communications. To make radios and wireless networks truly cognitive, however, is by no means a simple task, and it requires collaborative effort from various research communities, including communications theory, networking engineering, signal processing, game theory, software-hardware joint design, and reconfigurable antenna and radio-frequency design. In this paper, they provide a systematic overview on cognitive radio networking and communications by looking at the key functions of the physical, medium access control (MAC), and network layers involved in a cognitive radio design and how these layers are crossly related. In particular, for the physical layer, they are addressing signal processing techniques for spectrum sensing, cooperative spectrum sensing, and transceiver design for cognitive spectrum access. For the MAC layer, they review sensing scheduling schemes, sensing-access tradeoff design, spectrum-aware access MAC, and cognitive radio MAC protocols. In the network layer, cognitive radio network tomography, spectrum-aware routing, and quality-of-service control will be addressed. Emerging cognitive radio networks that are actively developed by various standardization committees and spectrum-sharing economics will also be

reviewed. Finally, the paper points out several open questions and challenges that are related to the cognitive radio network design.

DSA is a new spectrum sharing paradigm that allows secondary users to access the abundant spectrum holes or white spaces in the licensed spectrum bands. This concept is well described in [19]. DSA is a promising technology to alleviate the spectrum scarcity problem and increase spectrum utilization. While DSA has attracted many research efforts recently, in this article, they discuss the challenges of DSA and aim to shed light on its future. They first introduce the state-of- the-art in spectrum sensing and spectrum sharing. Then, examine the challenges that prevent DSA from major commercial deployment. The paper emphasizes that, to address these challenges, a new DSA model is critical, where the licensed users cooperate in DSA and hence much more flexible spectrum sharing is possible. Furthermore, the future DSA model should consider the political, social, economic, and technological factors all together, to pave the way for the commercial success of DSA. To support this future DSA model, the future cognitive radio is expected to have additional components and capabilities, to enforce policy, provide incentive and coexistence mechanisms, etc. The paper mentions the future cognitive radio with the expanded capabilities a network radio and discuss its architecture as well as the design issues for future DSA.

In [20], authors consider a communication system whereby multiple users share a common frequency band and must choose their transmit power spectral densities dynamically in response to physical channel conditions. Due to co-channel interference, the achievable data rate of each user depends on not only the power spectral density of its own, but also those of others in the system. Given any channel condition and assuming Gaussian signaling, they consider the problem to jointly determine all users' power spectral densities so as to maximize a system-wide utility function (e.g., weighted sum-rate of all users), subject to individual power constraints. For the discretized version of this non-convex problem, they characterize its computational complexity by establishing the NP-hardness under various practical settings and identify subclasses of the problem

that are solvable in polynomial time. Moreover, they consider the Lagrangian dual relaxation of this non-convex problem.

2.2 Blockchain

Blockchain [7], [21] is a distributed ledger that combines public-key cryptography, hashing, and distributed consensus to provide persistence, immutable, consistent, and highly available record keeping of transactions. These properties enable the disintermediation of trusted third-parties, prevent double spending of resources, and enhance public auditability. A set of special nodes in the network, called *miners*, are entrusted to build new pieces of the ledger (named *blocks*) while validating the transactions. A distributed consensus protocol is used to choose a miner to build a new block and the rest of the network believes the block created by the chosen miner as the ground truth. Therefore, miner selection and block creation should be sufficiently random to prevent gaming of the system while incentivizing many nodes to participate in the mining process. For example, Proof of Work (PoW) based consensus protocols choose the first node that produces a hash value of all the transactions in the block with a specific property, as the miner. Calculating such a hash is computationally expensive; hence, prevents an attacker from trying to alter the ledger state by modifying a series of blocks in the presence of a majority of the honest miners who compete to build the next block. Miners are rewarded with cryptocurrency for expending resources to build new blocks. Proof of Stake (PoS) and Proof of Authority (PoA) are alternative consensus protocols which are faster and resource efficient. However, public blockchain networks tend to prefer PoW over other consensus protocols due to its difficulty to game the system.

Smart contracts are a self-executing piece of code that is used to update the ledger state under a complex set of conditions. They are self-enforcing to facilitate contractual clauses digitally. Smart contracts are stored and executed within the blockchain nodes making them immutable. Smart contracts are written in specialized programming languages, e.g., a Turing-complete programming language called Solidity [8] is used in Ethereum [15].

Thus, it is equipped with sophisticated features for smart contracts where a smart contract can be in different states more than just two, unlike in Bitcoin [21]. This state fullness of smart contract in Ethereum is because it has a persistent storage space in blockchain; a key/value store which can be access by operations. A predefined set of conditions are checked before the execution of the smart contract and if any fails, the execution terminates, and changes revert as in atomic transactions.

Recently there have been developed a lot of applications employing blockchain, basically to facilitate secure and private final transactions. Blockchain acts as a universal spreadsheet which removes a central point of system failure and have been proposed as a way to secure online transactions.

The main features of blockchains are:

- Distributed - Because blockchain is a distributed database, there is no centralized copy. It makes the system robust against hacks.
- Secure - The distributed database is encrypted by private and public keys.
- Public - There is no central authority to validate or record transactions in a blockchain which leads to a more transparent system with security.
- Permission less - Because there is no single trusted user as the central authority, applications can be added to the overall system without seeking the approval of other users.

The placing of a transaction in a block is called a successful conclusion to a proof of work challenge and is carried out by special nodes called miners. Proof of Work is a system that requires some work from the service requester, usually meaning processing time by a computer. Producing a proof of work is a random process with low probability, so normally a lot of trial and error is required for a valid proof of work to be generated. If a miner produces a block that is approved by an electronic consensus of nodes, then the miner is rewarded with coins. Figure 2.1 describes how a blockchain network works with the initiation of a transaction.

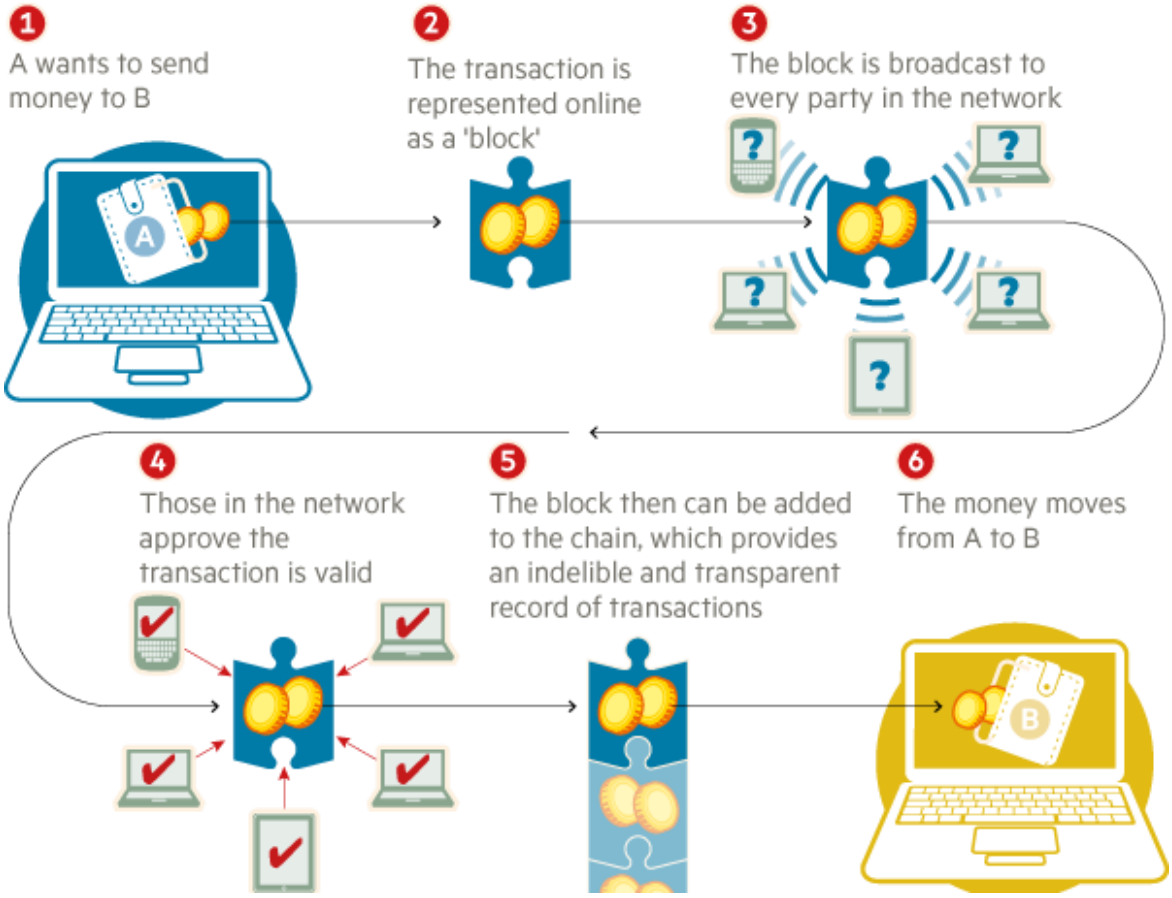


Figure 2.1: How Blockchain works

Bitcoin proposed by Satoshi [21] combines a simple decentralized consensus protocol, based on nodes combining transactions into a *block* every ten minutes creating an ever-growing blockchain, with proof of work as a mechanism through which nodes gain the right to participate in the system. While nodes with a large amount of computational power do have proportionately greater influence, coming up with more computational power than the entire network combined is much harder than simulating a million nodes. The double-spending problem is solved using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of

attacker nodes. To accomplish this without a trusted party, transactions must be publicly announced, and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found, and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one. New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

The reward is not the only incentive for miners to keep running their hardware. The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation Free. The incentive may help encourage nodes to stay honest. If a

greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favor him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

Three approaches are used in building advanced applications on top of cryptocurrency: building a new blockchain, using scripting on top of Bitcoin, and building a meta-protocol on top of Bitcoin. Building a new blockchain allows for unlimited freedom in building a feature set, but at the cost of development time and bootstrapping effort. Using scripting is easy to implement and standardize, but is very limited in its capabilities, and meta-protocols, while easy, suffer from faults in scalability. With Ethereum, we intend to build a generalized framework that can provide the advantages of all three paradigms at the same time. The intent of Ethereum is to merge together and improve upon the concepts of scripting, altcoins and on-chain meta-protocols, and allow developers to create arbitrary consensus-based applications that have the scalability, standardization, feature-completeness, ease of development and interoperability offered by these different paradigms all at the same time. Ethereum does this by building what is essentially. The ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. A bare-bones version of Namecoin can be written in two lines of code, and other protocols like currencies and reputation systems can be built in under twenty. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of our platform, with vastly more power than that offered by bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and state.

2.3 Blockchain and Spectrum Management

These features of blockchain and smart contracts suit DSA because primary users can share spectrum with secondary users by agreeing on terms and fees using a smart contract without the involvement of a trusted third party. In the proposed system, unauthorized users cannot get the license to access spectrum without making the necessary payment to licensed user. This should not be confused with malicious users who intentionally jam a particular frequency. There is nothing preventing physical radio interference if a malicious third party wants to wreak havoc on the system. A smart contract-based solution is optimal when all parties play nice and want to collaborate to maximize efficiency and usability of the available spectrum. Consequently, spectrum sensors that monitor use of the spectrum which is connected to regulators and law enforcement are still necessary for the correct operation of the smart contracts. Also, smart contracts could ensure a primary user cannot hold on to the license after agreeing to share spectrum or lease it to another secondary user at a larger amount while a contract is valid. These characteristics guarantee the security of the system provided all parties collaborate and there is no intentional malicious activity at the radio frequency level. As the blockchain is decentralized, it prevents single point failures while increasing the transparency of the system and enhances auditability of transactions to track unauthorized spectrum access.

In [13] they propose a blockchain verification protocol as a method for enabling and securing spectrum sharing in cognitive radio networks. The spectrum sharing mechanism is used as a medium access protocol for accessing wireless bandwidth among competing cognitive radios. They have introduced a virtual currency, called “Specoins”, for payment to access spectrum. An auction mechanism based on a first-come-first-served queue is used, with the price for spectrum advertised by each primary user in a decentralized fashion. The blockchain protocol facilitates the transactions between primary and secondary users and is used to validate and save each user’s virtual wallet. The blockchain also serves as a distributed database that is visible by all participating parties and any node can volunteer to update the blockchain. The volunteer nodes are called miners and they are awarded with Specoins. They have proposed diverse methods to exchange

the Specoins in order to make leasing possible even by cognitive radios that are not miners. The improvement of the proposed algorithm is shown compared to the conventional Aloha medium access protocol in terms of spectrum usage.

In [12], mobile network operators can expand their capacity by aggregating their licensed spectrum with the spectrum discovered opportunistically, i.e., spatiotemporally unused spectrum by other primary users. For an accurate identification of the spectral opportunities, the mobile network has to deploy multiple sensors, or it can offload this task to nearby nodes with sensing capabilities, so called helpers. Unfortunately, incentives are limited for helpers to perform energy-wasteful spectrum sensing. Instead, they envision spectrum sensing as a service (Spass) in which a smart contract running on a blockchain describes the required sensing service parameters and the contracted helpers receive payments only if they perform sensing accurately as agreed in the contract. In this paper, they first introduce Spass and derive a closed formula defining the profitability of a Spass based business as a function of the spectral efficiency, cost of helpers, and cost of the service. Moreover, they propose two-threshold based voting (TTBV) algorithm to ensure that the fraudulent helpers are excluded from Spass. Via numerical analysis, they show that TTBV causes almost zero false alarms and can exclude malicious users from the contract after only a few iterations. Finally, they have developed a running prototype of Spass on Ethereum blockchain.

As mentioned in paper [10] the Blockchain technology has received religious attention in the financial and internet domains, and recently interest has spread to adjacent sectors like communications. The paper [10] seeks to identify the impact of the blockchain technology in novel spectrum sharing concepts using the Citizens Broadband Radio Service (CBRS) concept as an example. The results indicate that the blockchain core characteristics can be utilized in several use cases addressing current CBRS implementation considerations. The CBRS concept could particularly benefit of blockchain's in building trust, consensus and lowering the transaction cost. In blockchain deployments, confidentiality should be taken into consideration through hybrid and private blockchain options. Furthermore, the

cognitive radio spectrum sharing blockchain combination paves the way for new business models and distributed services.

2.4 Digital Token

In [22] the authors discuss several uses of blockchain and, more generally, distributed ledger technologies outside of cryptocurrencies. They take a pragmatic view, focusing on three main areas: the role of coin economies for "data malls" (specialized data marketplaces), data provenance (a historical record of data and its origins), and "keyless payments," which are payments that can be made without having to know other users' cryptographic keys. They also discuss voting and other areas and give a sizable list of academic and nonacademic references using the digital tokens.

In [23], authors provide an overview of key economic aspects surrounding Initial Token Offerings (ITOs) and the issued application tokens (aka., appcoins). The purpose here is to layout a simple and accessible mental model of the structural and dynamic properties of ITOs and the overall market. It introduces about tokens, ITOs and the concept of regulatory arbitrage and how the latter may apply to the current token issuance process. A mental model is constructed to understand the structural (or microeconomic) properties of application tokens, and their proposed commercial purposes and issuance models. Key aspects of this analysis are to demonstrate the economic realities (and complexities) of the issuance process and the tokens themselves. Also, the ITO market dynamics (or macroeconomic) properties are assessed. Here the key focus is to highlight the potentially unsustainable and unstable state of the current market using known economic frameworks. A summary of the key legal, regulatory and communications aspects that could provide stability (or a softer landing) to the ITO market in a possible abrupt decline from the current state.

3 METHODOLOGY

3.1 System Overview

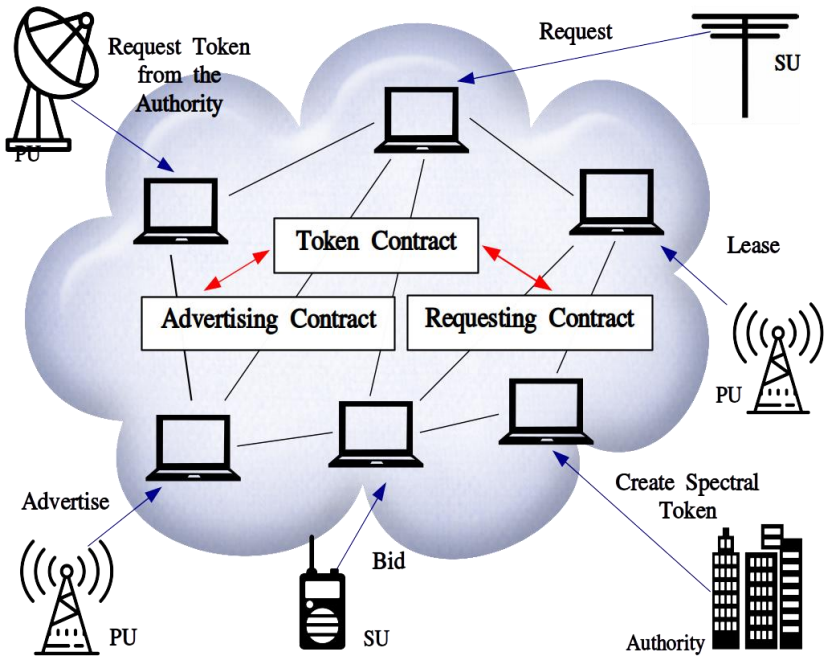


Figure 3.1: System overview - spectrum sharing among primary users, secondary users, and authority

We propose a DSA platform that enables spectrum utilization through a token-based approach involving the spectrum users who are willing to share the spectrum access in a decentralized manner. Let us consider the simplified example illustrated in Figure 3.1 to demonstrate key concepts of the proposed solution which is the system overview of the project. The system has three main end-users namely the authority, primary users, and secondary users. These end users can be either a person or a transmitting device like a cognitive radio. Regulatory bodies such as the FCC and TRC are considered as the authorities of the spectrum. The platform allows users to get ownership of frequency bands by paying to the authority. Ownership of each frequency band is coded as a digital token named *spectral token*. A spectral token resembles the license to access a particular frequency band within a specific geography and is signed with the private key of the

authority. This ensures that only the approved authorities issue valid spectral tokens. Once a spectral token is obtained the corresponding primary user may use the frequency band for its communication or lease it to secondary users. primary users may advertise the availability of frequency bands for lease to potential secondary users. This primary-initiated mode of operation is referred to as *advertising-based DSA.* Alternatively, secondary users may sense the spectrum using physical spectrum sensors which can provide feedback on what frequency bands are currently being exploited at the electromagnetic level while identifying the free frequency bands. Once identified, secondary users may request owners of spectrum holes to lease the spectrum via the proposed platform. This secondary-initiated mode of operation is referred to as *sensing-based DSA.* Depending on the initiator, location, time frame, and a fee to access a frequency band can be defined. Therefore, the proposed platform could conveniently support spectrum transactions of PAL users in CBRS [10] and spectrum users in IEEE 802.22 WRAN. An interface is also provided to the authority to validate the spectrum ownership at any time.

3.2 System Design

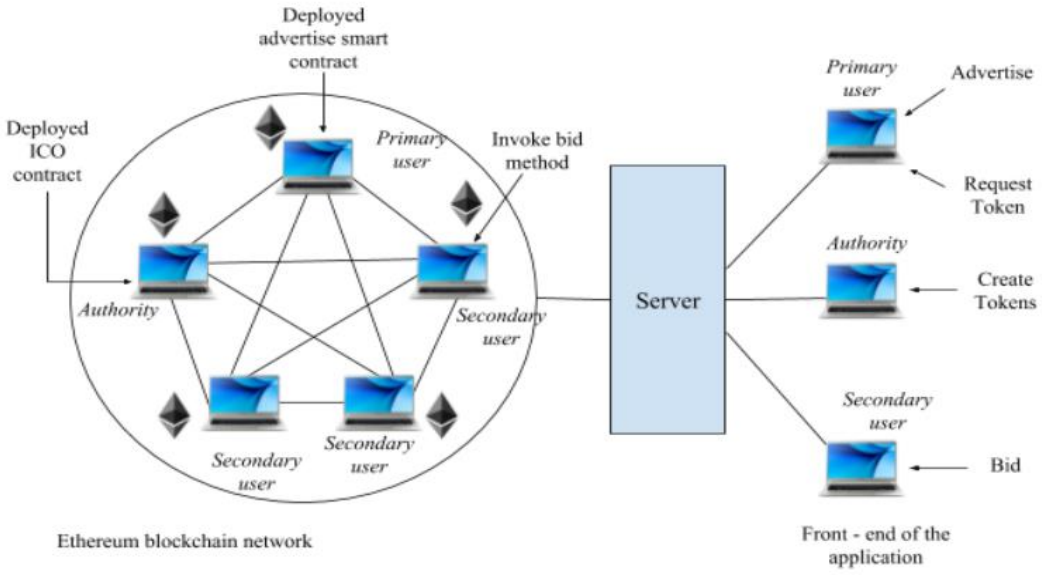
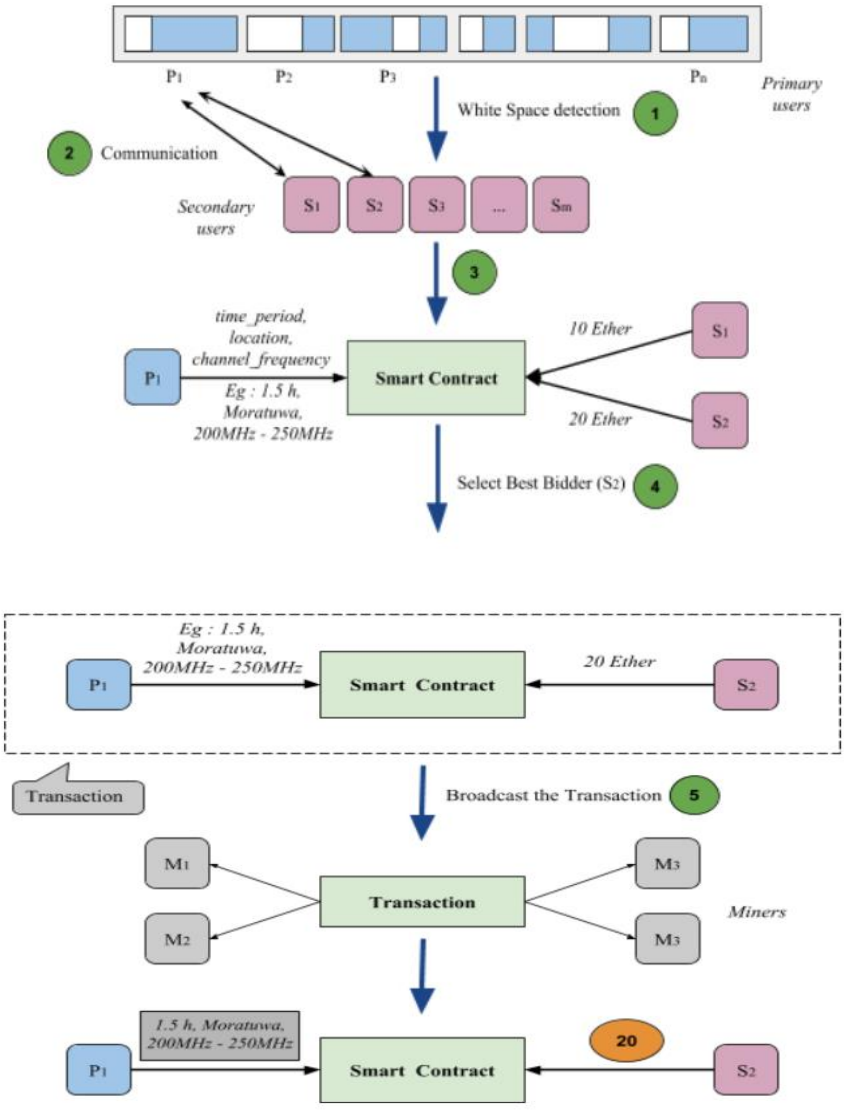


Figure 3.2: System Design

Figure 3.2 is the system design of the proposed platform. It basically reflects architecture of a decentralized application. The blockchain network is accessed by end users who are primary and secondary users through the front end of the decentralized application. They can advertise, request and bid for a frequency channel using the front-end user interface of the application. Here, involving a server is not a single point failure because all the data is secured in the blockchain. Something happening to the server would not impact the system because it is just a matter of bringing up a new server.



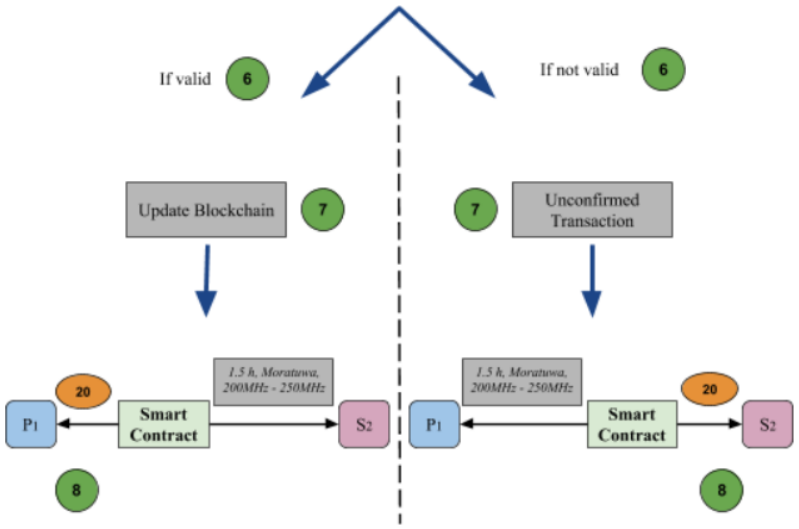


Figure 3.3: Proposed solution

Figure 3.3 shows the basic functionality of the proposed system. Initially the primary users who are the rightful owners of the spectrum get the ownership of the spectrum by paying for the frequency channels they need for, from the respective authority by which the spectrum regulated. Then they become the legal owners of the respective frequency channel.

The secondary users are the users who do not own a frequency channel but are in need of a frequency channel for communication and other purposes. They may not own a frequency channel either because they cannot afford a frequency channel or the whole spectrum for that location may be already allocated and sold for various parties. Since the primary users do not use all the frequency channels owned by them all the time, it causes vacant frequency holes in the spectrum and they can lend these white spaces to the secondary users who are in need of them.

First, the primary users as well as the secondary users should identify the available frequency channels (white spaces). This can be done by several methods; the primary user can advertise the available frequency channels, or the secondary user can sense the unused frequency channels.

Once the white space detection is done, the secondary user requests for the frequency channel in which he is interested in. There may be many secondary users who are interested in the same frequency channel in the same location for a particular time period. In order to eliminate this confusion, we have introduced the bidding concept to select a secondary user to which the primary user lends the respective frequency channel to. The selection of the best bidder is done by a smart contract. This approach adheres to the objective of developing a bidding platform for the negotiation between the primary user and the secondary user while identifying the varying bidding and pricing rule.

Once the best bidder is selected, it is necessary to check whether the bidder has enough currency in order to pay for the frequency channel which the primary user is hoping to lend and whether the respective frequency channel exists. Until this validation takes place, the smart contract holds on to the assets which are needed for the transaction (ether and the frequency channel which is represented by a spectral token). The validity (Check whether the secondary user has enough ether to pay and the primary has a valid frequency channel) of the transaction is checked by the blockchain by the mining process conducted in it. Since the blockchain handles the transaction so that no one can cheat, we don't have to trust a third party during the transaction process. Use of blockchain as the medium of conducting the transaction helps us to achieve our objective of making the transactions between primary and secondary users fast and secure without the involvement of a third-party.

Once the mining is done and the blockchain finds out that the transaction is invalid, the asset transferring between the primary and the secondary does not happen. If the transaction is valid, the asset transferring between the primary user and the secondary user takes place. This transaction ensures that the spectral token is lease for the time period specified in the smart contract and primary user get the ownership of the spectral token after leasing time is over.

Now let's look into design of major system components of the proposed system.

3.2.1 Spectral Token

It is necessary to ensure that interference is avoided as much as possible to guarantee effective data transmission by enforcing sequential access to a particular frequency band. A token-based access protocol in which a token resembles the license to use a frequency band would accomplish this requirement. Sequential access is guaranteed as only a single user can hold the token at a time. Therefore, token is one of the major components of the proposed system. Thus, we propose a crypto-token-based approach to facilitate the efficient use of the radio spectrum while minimizing interference.

Crypto-token is a virtual currency token that represents an asset or utility on a blockchain platform. Crypto tokens are of two types namely, utility and security. A *utility token* could be used to purchase goods or services offered by the members of the platform while a *security token* indicates the ownership of an underlying real-world asset. We propose to use a security token named *spectral token* to represent a frequency band and the entity that has the license to use that frequency band. These tokens are issued by a token contract that can only be initiated by the authority. Moreover, once a token is created, it can be altered or destroyed only by the authority. Each frequency band is uniquely identified as a contiguous range of frequencies (*startFreq* to *endFreq*) and its *location*. Here, a frequency band reflects the lowest frequency range that can be given out of the spectrum depending on the radio frequency region and anticipated applications. For example, if the lowest assigned frequency range is 5 MHz, a primary user purchasing a license for a 10 MHz band will get two 5 MHz spectral tokens for adjacent frequency bands issued by the authority. In future, if 1 MHz frequency bands are to be given out instead of 5 MHz, the authority may reclaim the 5 MHz spectral token and issue five new token for each 1 MHz frequency band. Depending on how a cell is defined, the location could be specified as an ID; name of a city, or a polygon. Every spectral token is bound to the licensed primary user and signed by the authority using its private key. Authority's signature is indicated as the issuer of the token. Therefore, primary users and secondary users can verify the validity of the spectral token using the public key of the authority which is known to

everyone. Additional attributes such as the transmit power level and other parameters relevant to the proper use of the frequency band can also be included in the token.

When a frequency band is leased, the ownership of the spectral token is transferred from the primary user to the secondary user using a smart contract. Thus, facilitating the secondary users to transmit via the spectrum legally. Once a user gets the ownership of a token, he/she can use the frequency band resembled by the token for the period agreed between the primary user and secondary user. Ownership of a spectral token is transferred via a token contract (i.e., a special form of a smart contract used to transfer crypto tokens). If the token is not yet leased, the ownership can only be transferred by the primary user to a secondary user. If it is leased, the secondary user can only transfer the token to the primary user (secondary user cannot transfer the token to other users). This prevents further leasing of license by secondary users. Once leased, only the intended secondary user can legally utilize the frequency band, as written into the smart contract by the primary user who owns that spectrum. After the termination of the agreed lease period, the spectral token needs to be automatically returned to the primary user before the termination of the smart contract. A spectral token can be primarily in two states. If T_i is a spectral token resembling *i*-th frequency band its state S_{T_i}; can be characterized by the following step function shown in equation 3.1:

$$S_{Ti} = \begin{cases} leased & owner \neq primary\_user \\ not\_leased & owner = primary\_user \end{cases}$$

Equation 3.1 : States of Spectral Token

Because the individual tokens should differ from each other, spectral tokens should be non-fungible. For example, ERC20 token standard [24] used on the Ethereum blockchain cannot distinguish the partitions of the spectrum separately because they are created and used in bulk. Whereas ERC721 is a non-fungible token standard [25] that facilitates the necessary features of spectrum sharing by creating unique tokens for each (*startFreq*, *endFreq*, *location*, *primary user*) tuple. Also. ERC721 token standard allows

to create tokens on demand. Therefore, the authority needs to create Spectral Token only

```
changeOwner (oldOwner, newOwner, tokenId) {
      if ((not leased and oldOwner = primaryUser) or
            (leased and newOwner = primaryUser)) {
            addToWallet (newOwner, tokenId);
            removeFromWallet (oldOwner, tokenId);
            Return true;
      }
      Return false;
}
```

Figure 3.4: Change owner method of Spectral Token

when the primary user requests for a respective frequency band. Therefore, we use the ERC721 standard to represent the spectrum license.

The pseudocode shown in Figure 3.4 illustrates the way the owner of the Spectral Token is changed. Initially the change owner function checks whether the change owner function can be executed, i.e. to prevent further leasing by secondary user. If the secondary tries to release this function does not execute. Once the this is verified, the respective token is added to the wallet of the *newOwner* and it is being removed from the wallet of the *oldOwner*.

3.2.2 Initial Spectrum Allocation

Governments regulate access to the spectrum via an entity such as the FCC or TRC. Such regulatory bodies are referred to as the *authority* in the proposed system. Tokens are created on the blockchain using a token contract which is popularly known as the ITO contract. ITO contracts are deployed on the blockchain to create digital tokens and manage their ownership. Therefore, this contract should be deployed at the time of system deployment by authority.

This mainly concentrates on issue of the Spectral Tokens to the Primary Users. The Spectral Token is issued by the authority only when a primary user request for it. Here, many primary users can request for the same frequency band, and authority give the

ownership to one of the primary users after checking the necessary terms and conditions that an owner of a token should have. If the authority is not satisfied with the characteristics of the primary user who requested for the token, the authority has the ability to decline the request. Once the token is created on demand, the primary user becomes the rightful owner of the spectrum and he/she gets the ability to lease the respective frequency band to the secondary users. During the creation of the spectral token, the ITO contract calls the "create_token" method in the "SpectralToken" contract. Primary users can request spectral tokens from the authority for unallocated frequency bands by completing the payment either through the platform using cryptocurrency or externally. In cases where frequency bands are already licensed, the authority could issue spectral tokens for the primary users of already assigned frequency bands at the time of deploying the proposed platform. The authority then generates a spectral token(s) by calling the *create_token* method in ITO contract. Finally, the ownership of the particular spectral token is transferred to the respective primary user. Subsequently, primary user can transfer the license to use the frequency band by transferring the ownership of the token to an interested secondary user by calling the *transfer_owner* method in ITO contract.

The pseudocode in Figure 3.5 explains the basic functionalities of an ITO contract. Once a primary user needs to get the ownership of a frequency band, the request method is called, and the details are added to the *spectrumRequestArray*. Once the authority accepts the request, the *create_token* method is called, and a token is created. At the meantime, the information stored about the respective request is deleted from the *spectrumRequestArray*. If the authority rejects the request from the primary user, no token is created, and the request is removed from the *spectrumRequestArray*.

```
function request (location, startFreq, endFreq) {
      spec.startFreq = startFreq;
      spec.endFreq = endFreq;
      spec.location = location;
      spec.primaryUser = msg.sender
      spectrumRequestArray.push(spec);
}

function remove_request(requestId) returns(bool) {
      if (msg.sender = authority){
            spectrumRequestArray.remove(spec[requestId]);
            Return true;
      }
      Return false;
}
function create_token (startFreq, endFreq, location, PrimaryUser) {
      SpectralToken.create_token (startFreq, endFreq, location,
                                          primaryUser);
}
```

Figure 3.5: Basic ITO contract

```
create_token {
    if (caller != authority) then return false
    if (token is already created) then return false
    i <- starFreq
    while(i < endFreq)
      Create a new Spectral token;
      Set tokenId, startFreq, endFreq, primaryUser,
                        location   of the SpectralToken;
      Set the signature of the authority issuer of SpectralToken;
      i <-  i + min_allocatable_size_of_a_freq_band
      Add to token list
}
```

Figure 3.6: Create token method of Spectral Token

The pseudo code in Figure 3.6 explains the *create_token* method in *SpectralToken* contract which is being called by the ITO contract. First, the method checks whether the authority is the one who is calling the method. If not, the token is not

created. Not only that the availability of the token is checked. I the token is already available the user is informed the availability of the particular token. Here, the necessary parameters for the spectral token which resembles a particular frequency band is set. Finally, the token is being added to the list of tokens issued by the authority.

3.2.3 Advertising-Based DSA Model

The Advertising based dynamic spectrum access model is one other main component of the system. Advertising is a way to enable the secondary user market for dynamic spectrum sharing. In this model, primary users of the frequency bands can advertise their frequency bands for secondary users specifying the time period for which the particular frequency band is vacant to use for data transmission. As this advertising process is done by primary users, they can guarantee that no incumbent usage will occur during the advertised period of time. Therefore, the main problem of dynamic spectrum access which is the interference would be minimized from this approach of advertising vacant frequency bands by primary users. On the other hand, primary users should have to have an incentive to lease their owned frequency bands while providing the license to secondary users to access the frequency band for a specific period. So, the primary users of the system should be able to specify a particular fee to allow access to secondary users.

The main concerns of this approach are of two perspectives. As this is an agreement between primary and secondary users, there are concerns from both the parties. First let's consider the concerns from primary user. Primary users are concerned about how to guarantee that the payment will be received for granting the access to their frequency band after the leasing agreement is placed, how to maximize the profit of leasing the frequency band and how to guarantee the transfer of the license back after the agreed time period is due. The concerns of the secondary users will be how to guarantee the advertised frequency bands are owned by existing primary users, how to confirm the transfer of the ability to access the spectrum within a particular time period and how can we guarantee the licensed time period will sustain without claiming of license. We address these

concerns in the proposed system. The agreement between primary user and the secondary user is interpreted in our system as a smart contract deployed in the blockchain. Smart contracts can digitally enforce the contractual clauses which should be considered before the transactions between primary and secondary users are initiated. Most of the time these smart contracts are reusable so that they can be used by multiple users in the system.

In this model, transactions are initiated by primary users who advertise the availability of frequency bands for leasing. A primary user advertises a frequency band by deploying a smart contract to the blockchain platform. The demand for frequency bands vary according to time, location and situation. Therefore, advertising parameters such as the bidding time, leasing time, and leasing policy should be customizable when coded into the smart contract. This customizability of smart contracts can manage the spectrum access for future time also which helps to utilize the spectrum more effectively. In the proposed platform we support FCFS and competitive bidding models. More complex agent-based negotiation models could be built as smart contracts. However, we differ such smart contracts for future work. If C_i is the *i*-th advertising smart contract deployed to lease a spectral token for a future time frame, its states S_{Ci} can be characterized by the following step function shown in equation 3.2:

$$S_{Ci} = \begin{cases} pending & \begin{cases} initiated & bidStartTime > currentTime \\ & bidStartTime < bidEndTime \\ & bidEndTime \leq lendStartTime \\ & lendStartTime < lendEndTime \\ bidding & bidStartTime \leq currentTime \\ & bidEndTime > currentTime \\ lended & lendStartTime \leq currentTime \\ & lendEndTime > currentTime \end{cases} \\ ended & lendEndTime < currentTime \end{cases}$$

Equation 3.2 : States of Advertising Smart Contract

The smart contract should be initiated only if the conditions for the initiated state are met. Otherwise the deployment of the smart contract should be rejected. The smart contract

should allow secondary users to bid for it, only in the state of *bidding*. The ownership of the frequency channel must only be transferred when the smart contract in *lended* state. In all these states the smart contract is in pending state and finally the smart contract is *ended* when leasing time is expired, and ownership of the frequency band can be transferred back to the primary user.

Figure 3.7 and Figure 3.8 shows the pseudocode of an advertising smart contract which contains the basic functionality for the advertising smart contract. Advertising parameters can be set using the *advertise* method which is invoked at the time of smart contract deployment. Such customizability enables primary users to advertise frequency bands to lease immediately or for a future time frame. Once the frequency band is advertised, secondary users should be able to search for the frequency bands advertised for a given location. Potential secondary users should be able to access parameters such as bandwidth, leasing time, and maximum bids of frequency bands available for the searched location. As smart contracts are deployed per advertising and they hold different contract addresses. Therefore, searching advertising contracts are bit challenging. Therefore, we trigger an event of a common contract deployed in the blockchain when every smart contract is deployed so that parameters could be retrieved by querying the blockchain transactions. Advertised frequency bands are indexed based on the location so that they can be searched efficiently. If the leasing policy of a particular advertised frequency band is competitive bidding, multiple secondary users can bid for the frequency band using the native cryptocurrency of the blockchain which is in our case Ether during the bidding time specified in the smart contract. This is achieved by invoking the *bid* method in the pseudocode. Secondary users can see the maximum bid offered so far and alter their bids to a higher value within the bidding time enabling competitive bidding. Here all the bids from secondary users are held by the respective smart contract.

```
Contract Advertise //deployed by primary user
//called by primary user at the time of deployment of the contract
advertise {
   if (msg.sender != primary_user of tokenId) or
                  state(contract) != initiated
   else Set token_id, minimum_bid, bidding_start_time,
            bidding_end_time, lend_start_time, lend_expiration,
            maximum_bid <- 0, best_bidder <-null
}
//called by secondary users
bid {
   if (msg.sender is contract_owner) or
                  state(contract) != bidding or
                  (minimum_bid > bidding_value) then
            return false
   Add msg.sender as a bidder and map the bidding value for
                                          corresponding bidder
   if maximum_bid < bidding_value then
            maximum_bid <- bidding_value
            best_bidder <- msg.sender
            return true
}
```

Figure 3.7: Basic pseudo code for Advertising contract 1

```
//called by primary user at lending start time
transfer_token {
    if (msg.sender != contract_owner) or
                    state(contract) != lended then
            return false
    if state(token_id) is leased then
            foreach bidder: bidder.transfer(corresponding bid)
            return false
    else transfer_owner(from: primary_user,
                                to: best_bidder, token_id)
            foreach bidder != best_bidder:
                                bidder.transfer(corresponding bid)
            return true
}
//called by primary user at lend expiration time
transfer_assets {
   if (msg.sender is not contract_owner) or
                    state(contract) != ended then
            return false
   transfer_owner(from: best_bidder, to: primary_user, token_id)
   primary_user.transfer(maximum_bid)
}
```

Figure 3.8: Basic pseudo code for Advertising contract 2

When the bidding time is over, smart contract selects the best bidder as the lessee. At the beginning of the agreed leasing time, the *transfer_token* method should be triggered to transfer the ownership of the token from primary user to secondary user who offered the maximum bid. As the transaction is signed using primary user's private key, secondary user can validate the transaction using primary user's public key. After the expiration of the leasing time, the *transfer_assets* method can be triggered to transfer the ownership of

the spectral token back to primary user. Agreed fee (i.e., maximum bid) for the use of the frequency band is charged from secondary user using the cryptocurrency native to the blockchain platform and transfer to the primary user's account. The transfer could happen as soon as a deal is made. Alternatively, it may first go to an escrow account on the blockchain platform and later transfer at the beginning or end of the agreed leasing time. This is necessary for CBRS like DSA, as the incumbent may suddenly request the frequency band for its use terminating the contract.

3.2.4 Sensing-Based DSA Model

The Sensing based dynamic spectrum access model is another component of the system which is somewhat similar to advertising model discussed in the previous section. Sensing is also a way to enable the secondary user market for dynamic spectrum sharing. In this model, secondary users of the system can detect vacant frequency bands (white spaces) by sensing the spectrum physically using radio frequency sensing devices. Then they can request vacant frequency bands from primary users of those frequency bands for a specific time period. We cannot guarantee that sensed frequency bands may be vacant for the whole time of period that secondary users are requesting. Therefore, to minimize the interference, acceptance for the request from the primary users are needed. So, transmitting data should be done by the secondary after they receive the consent from primary users. On the other hand, primary users should have to have an incentive to lease their owned frequency band while providing the license to secondary users to access the frequency band for the requesting time period. So, the secondary users should offer a bid for which they are requesting the channel. After that primary users can accept or decline the request from secondary users depending on the time duration, they are requesting frequency band.

The main concerns of this approach are also of two perspectives. As this is an agreement between primary and secondary users, there are concerns from both the parties. First let's consider the concerns from primary user. Primary users are concerned about how to

guarantee that the payment will be provided for granting the access to their frequency band after the leasing agreement is placed, how to maximize the profit of leasing the frequency band and how to guarantee the transfer of the license back after the agreed time period is due. The concerns from the secondary users will be how to identify the primary users of the vacant frequency bands, how to confirm the transfer of the ability to access the spectrum within a particular time period and how can we guarantee the licensed time period will sustain without claiming of license. So, we have addressed all these concerns in our proposed system design.

In this model secondary users sense for free frequency bands at a given location. Once a free frequency band(s) is identified, a potential secondary user could request the necessary frequency band(s) from primary user(s) through the proposed platform. This is achieved by invoking the sensing smart contract on the blockchain which passes a message to the frequency band's owner while storing details of the request such as the address of secondary user, price he/she willing to pay, requested time duration and expire time of the request. The Figure 3.9 shows the pseudocode of the request smart contract which shows the basic functionality. Here, the secondary user can retrieve primary user's information from the token contract as they are stored in the corresponding smart contract during the token creation. If the primary user of the requested frequency band is connected to the system at the time the request is made, the system will notify the primary user by retrieving the information from blockchain. primary user could decide whether to lease the frequency band based on the availability during the requested time. If multiple secondary users request the same frequency band at the same time and primary user agrees to lease it, when primary user accept request in request smart contract by specifying the leasing time the smart contract will select the best offer and transfer the token. If the primary user is not willing to lease the requested frequency band, then the transaction terminates while notifying the secondary user about the denial of the request. Once the best offer is finalized, both the spectral token and assets are transferred similarly to that in the advertising-based mode.

```
Request Contract //deployed at the system deployment by authority

request() //called by secondary user at the time of identification
of free frequency bands
  if(now > exp_time) and state(token) == leased
      return false
  Set request parameters(token, msg.sender, exp_time, duration,
                                                        primary)
  Add to requestors and requests
  notify primary


accept() //called by primary at notification from secondary
  if (msg.seder is not primary user of the token) then return false
  else if (state(token) == leased) then return false
  Select best bidder and bid from requests with acceptable time
                                                        duration
  Set the status of the selected request to granted
  transfer_owner(token)
  msg.sender.transfer(best_bid)


decline() //called by primary at notification from secondary
  Check msg.seder is primary user of the token
  Set the status of the requests to declined
  Notify secondary users


withdrawFunds() //called by the secondary users after expiration or
decline of the request
  if (status(request) == declined or exp_time < now)
  msg.sender.transfer(bid)
```

Figure 3.9: Request contract pseudocode

4 IMPLEMENTATION

The PoC implementation of the proposed platform includes a front-end application, back-end application, and the blockchain. We implemented front-ends targeting both the mobile and web-based devices. Any radio frequency-enabled device could use these interfaces to request a license from the authority, advertise license to secondary users, request a license from a primary user, as well as track ownership of a license with time. A mobile application is implemented to demonstrate the end-to-end functionality of the system because sensing, transmitting, and creating a hotspot can be initiated through the mobile interface promptly. Here, the creation of the mobile hotspot in a particular radio frequency channel implies the transmission by the device (i.e., user) via the frequency band. Moreover, the back-end application which sits between the front-end and blockchain is used to query the blockchain using JSON based RPC calls as per the requests from the front-end. We choose Ethereum [15] blockchain to demonstrate the PoC solution, as it uses PoW for consensus, has a rich smart contract language, and supports non-fungible tokens. We set up a private Ethereum blockchain to deploy smart contracts without the need for the real cryptocurrency, as well as to control experimental parameters. The difficulty of block mining is set to a much lower value than in the public Ethereum main network to save computational power.

4.1 Frontend

The front-end application consists of web interface and mobile application interface. The web application is implemented using ReactJs and the mobile application is implemented using Android. End users of the system can access the interfaces using these interfaces and to do the spectrum transactions. These interfaces provide the easy deployment of the smart contracts in user friendly manner. Below shown are some of the screen captures from the front-end application.

4.1.1 Web Application

The web application can be used by the spectrum users (primary and secondary users) as well as the authority. Separate home pages are provided for the spectrum users and the authority.

Authority Section

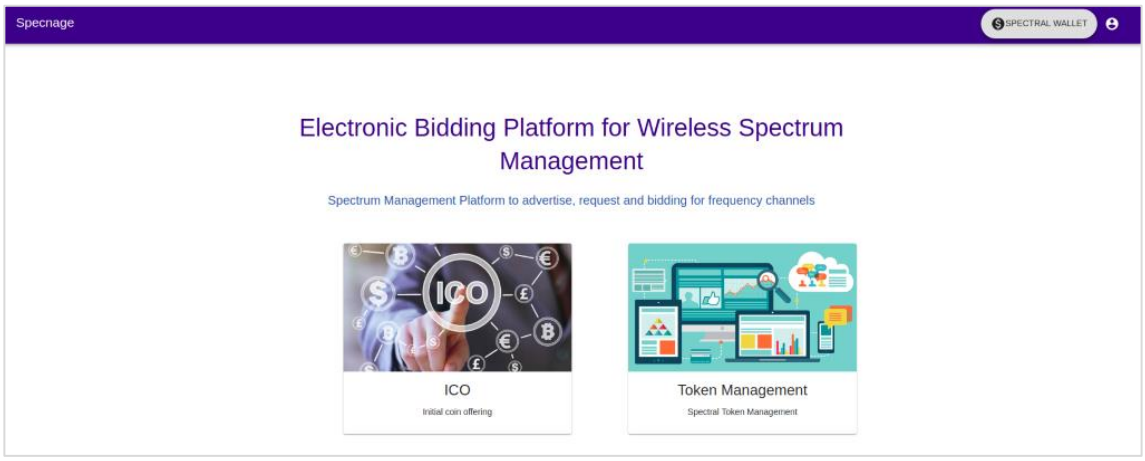


Figure 4.1: Authority Home page

From **Initial Token Offering**, authority can issue tokens for frequency bands that are requested by primary users.

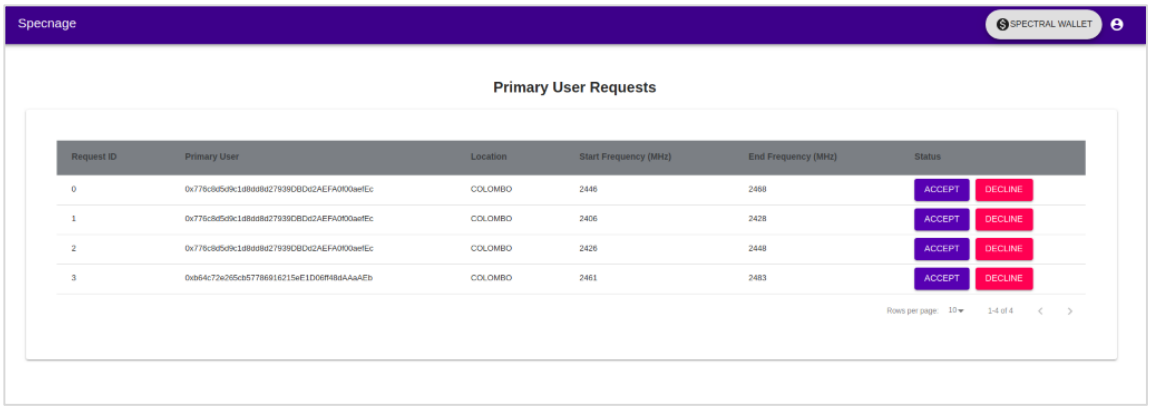


Figure 4.2: Initial Token Offering Interface

From the interface in Figure 4.2, authority can accept/reject primary user requests for a frequency band. If the request is accepted, requested user is granted the ownership of that token.

Then, from the **Token Management section**, authority can keep track of a particular spectral token by getting different information of a token.

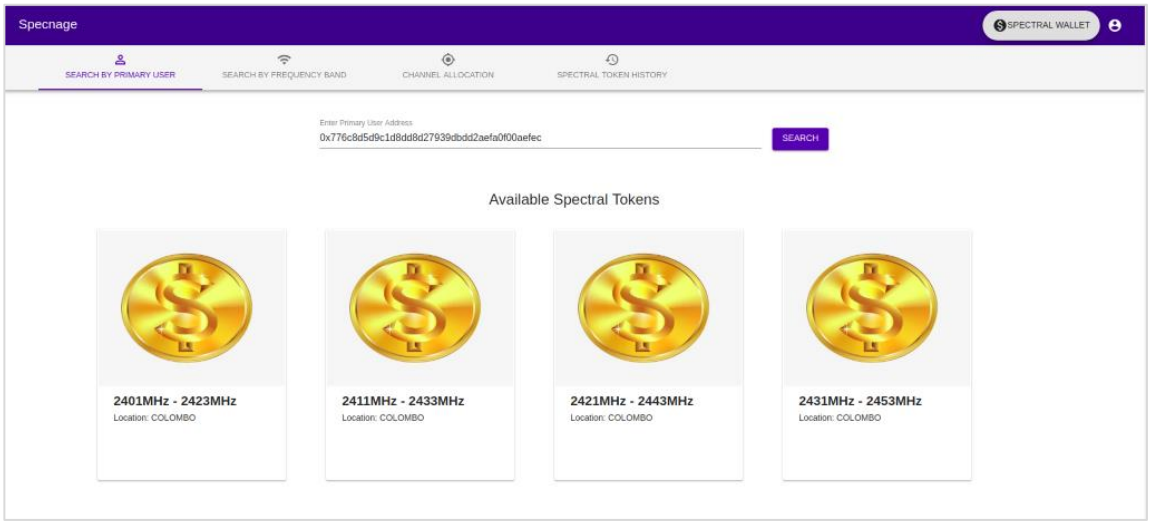


Figure 4.3: Search by primary user

Figure 4.3 shows the interface to get the spectral tokens owned by a particular primary user by giving primary user address.

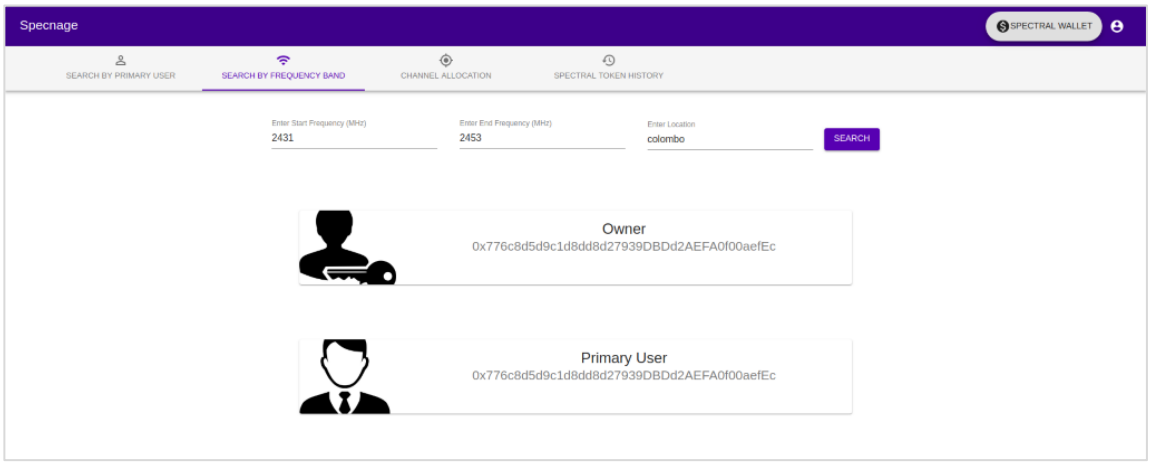


Figure 4.4: Search by frequency band

Figure 4.4 shows the interface to get the owner and primary user of a spectral token. Authority can find them by giving frequency band and the location of the desired spectral token.

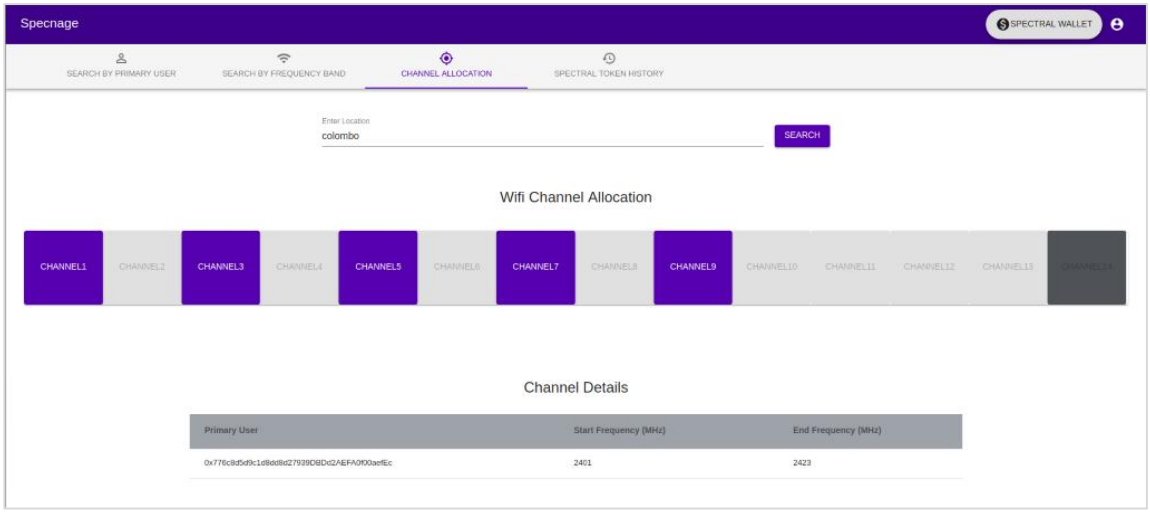


Figure 4.5: Wi-Fi Channel Allocation interface

Figure 4.5 shows the interface to get the channel allocation of Wi-Fi band by giving a location as the search key. Allocated frequency bands are indicated in blue color while unallocated are indicated in gray color. Once the user clicks on an allocated channel, he/she can find the primary user of that particular channel.

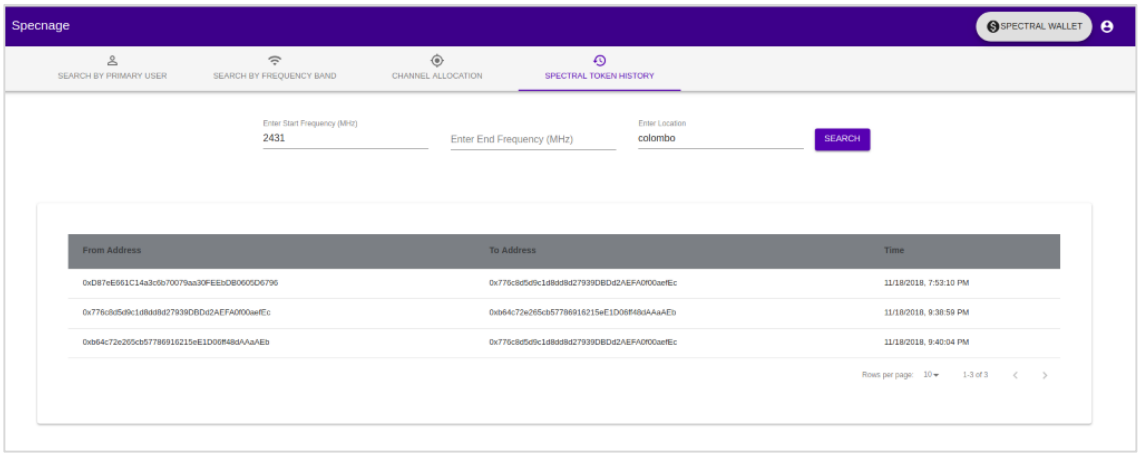


Figure 4.6: Spectral Token History interface

From the interface shown in Figure 4.6, authority can get the history of a particular token by providing channel frequency and location of the token. Form these details, authority can find the primary user of a token at a particular time and track the token ownership with time.

Spectrum Users Section

From the spectrum users' section, primary users can advertise their vacant frequency bands, secondary users can request tokens from the authority and secondary users can place their bids for an advertised frequency band.

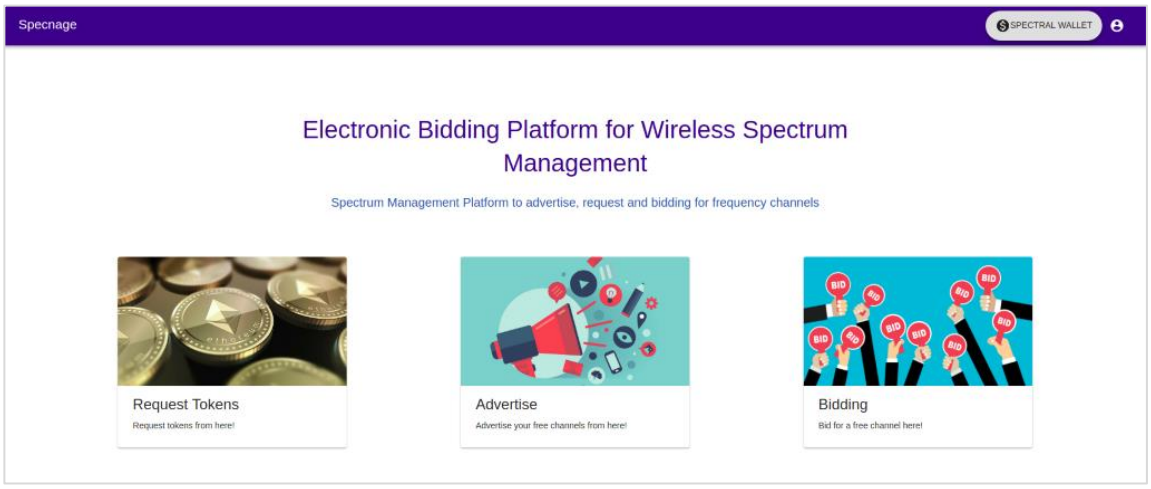


Figure 4.7: Spectrum User Home page

Once the user clicks on the “Advertise” section, user will be directed to the interface shown in figure 4.8.

From the interface shown in Figure 4.9, primary user can advertise his vacant frequency band to secondary users for now or future time periods as well. And also, he can choose FCFS or competitive bidding as bidding policy.

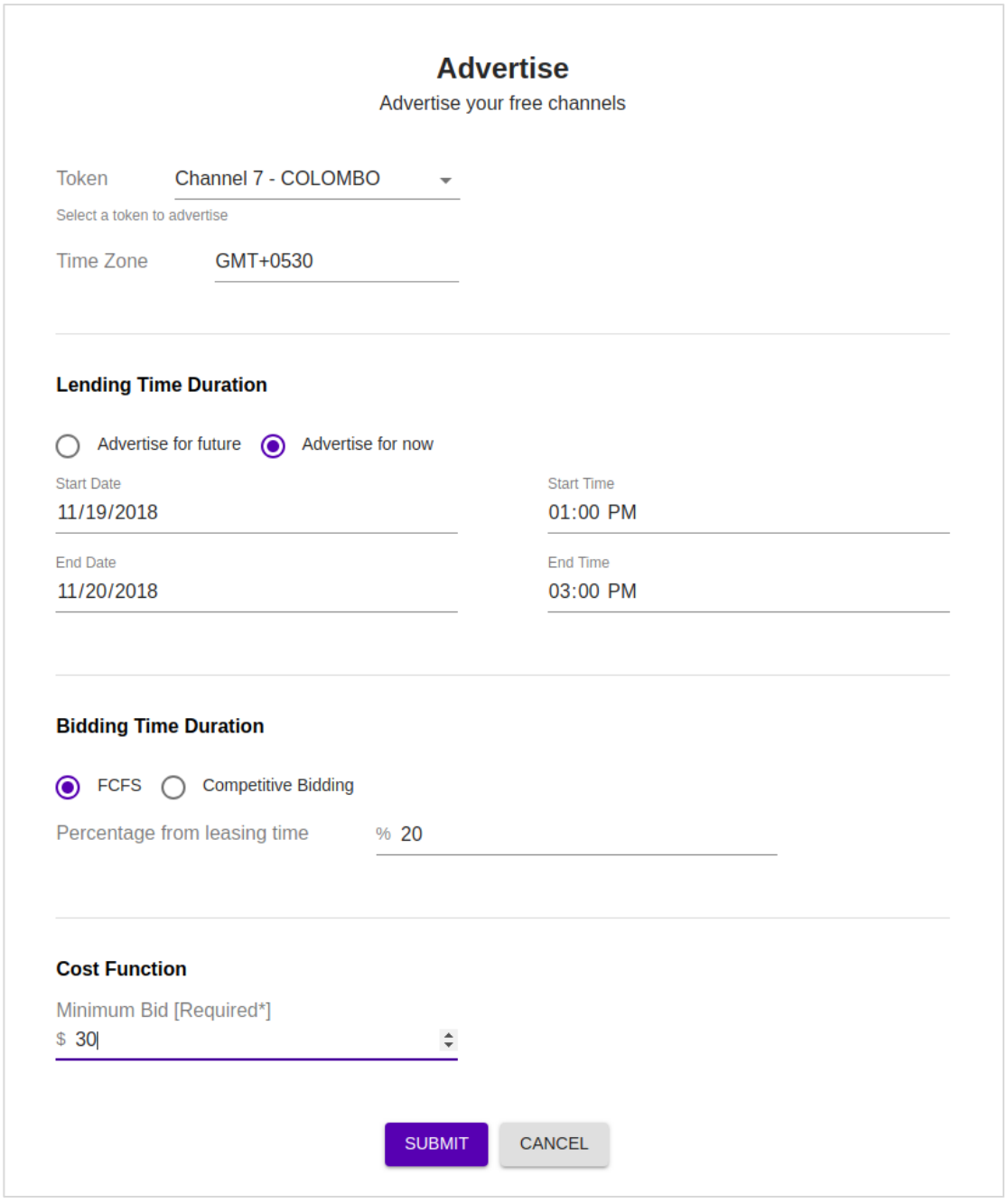


Figure 4.8: Advertise interface

Once the user clicks on "Bidding", he/she will be directed to the interface shown in Figure 4.9. User can get pending smart contracts of a given location. Then, he/she can place their bids for the interested frequency bands.

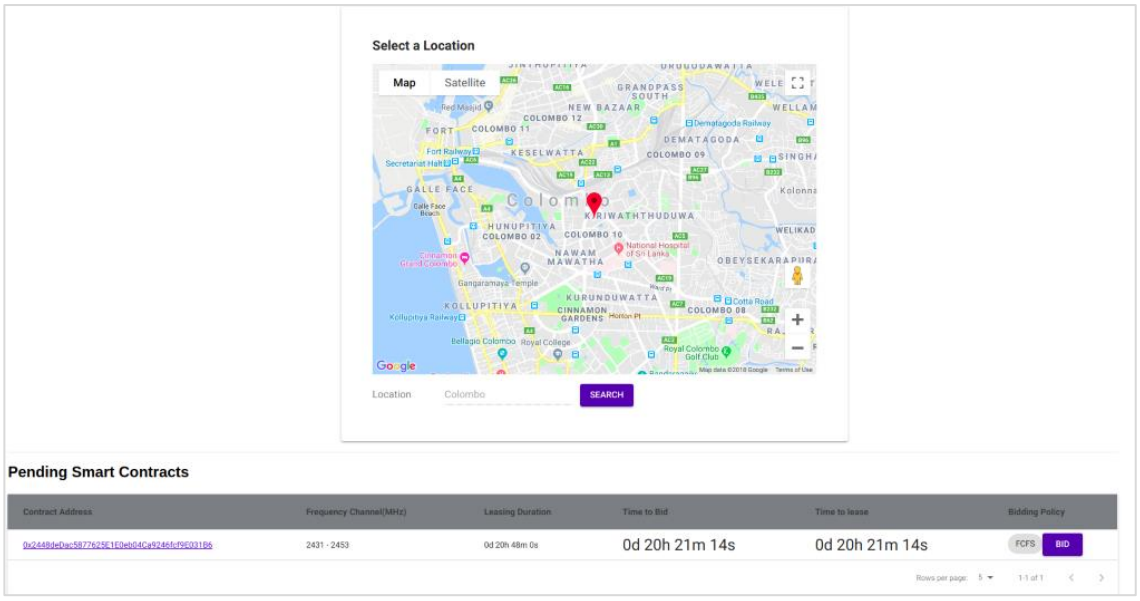
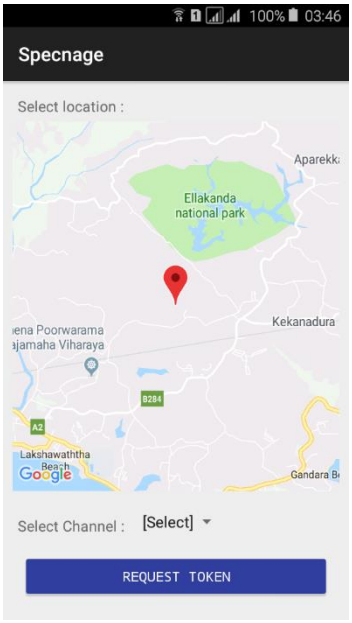


Figure 4.9: Advertised smart contracts

4.1.2 Mobile Application

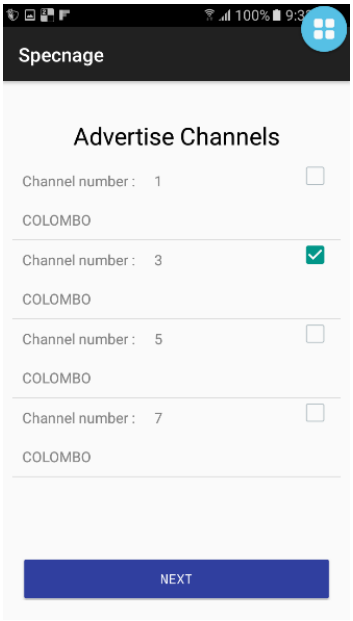
Step 1 - Request spectral token from the authority



From the interface in Figure 4.10 mobile user can request spectral tokens for a particular frequency band which are not already allocated in their preferred geographical area.

Figure 4.10: Request token interface

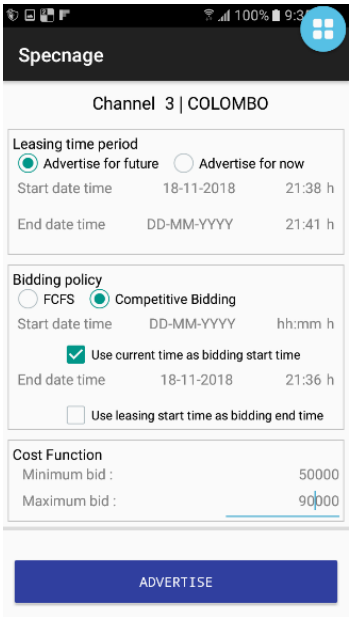
Step 2 - Display owned spectral tokens which are available for advertise



When the authority accepts the token request from a primary user, then details of that spectral token will be displayed in the interface in Figure 4.11. Primary user can advertise spectral tokens which are displayed here.

Figure 4.11: Interface of Advertisable channels

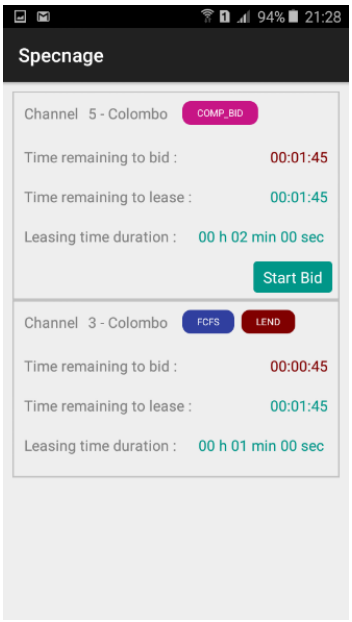
Step 3 - Advertise Spectral token



From the interface in Figure 4.12, primary user can advertise his vacant frequency band to secondary users for now or future time periods as well. And also, he can choose FCFS or competitive bidding as bidding policy.

Figure 4.12: Advertise interface

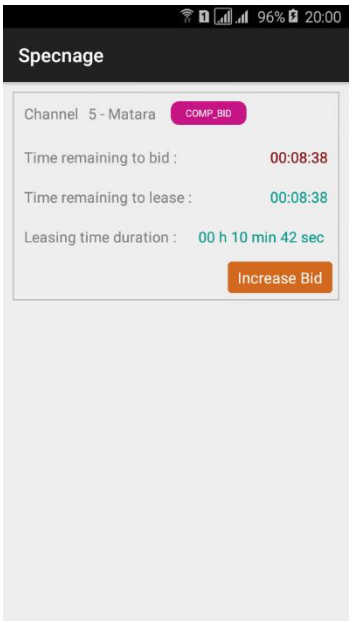
Step 4 - Display advertised contracts



The Figure 4.13 shows the interface which secondary users can view available live contracts they can bid. This interface shows details about each contract such as channel number (Start frequency and end frequency), location, bidding policy, leasing time duration, time remaining to bid and lease. From this interface secondary users can request frequency bands from primary users.

Figure 4.13: Advertised contracts

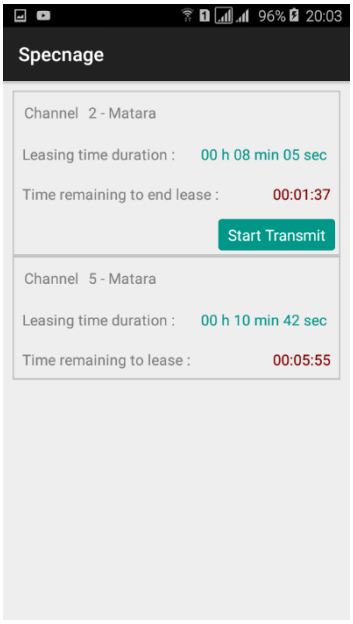
Step 5 - Increase bid by secondary users



In competitive bidding contracts bidders can increase their bidding value during bidding time duration. The interface in Figure 4.14 facilitates that functionality to the user.

Figure 4.14: Increase bid interface

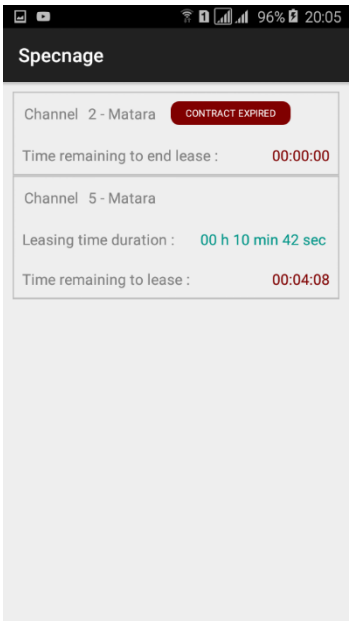
Step 6 - Transmit data using bought channel



If the contract is FCFS, secondary user who bid for that contract first will get that frequency band. But the transmitting time is started when the leasing time is started. If the contract is competitive bidding, blockchain selects the best bidder for a particular contract and allow that secondary user to transmit using the frequency band from the leasing start time.

Figure 4.15: Start transmit interface

Step 7 - Ownership transfer back to the Primary user



When the leasing time expires ownership will be transferred back to the primary user.

Figure 4.16: Expired contracts interface

4.2 Blockchain

We set up a private Ethereum blockchain to deploy smart contracts without the need for the real cryptocurrency, as well as to control experimental parameters.

We used Geth which is a command line interface (CLI) tool that communicates with the Ethereum network and acts as the link between the computer and the rest of the Ethereum nodes. To run a private network, provided Geth with some basic information required to create the initial block. Every blockchain starts with a Genesis Block**,** the very first block in the chain. To initiate the private blockchain, we created a genesis block with a custom genesis file. Then, ask Geth to use that genesis file to create our own genesis block.


```
{
    "nonce": "0x0000000000000042",
    "timestamp": "0x0",
    "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "extraData": "0x00",
    "gasLimit": "0x8000000",
    "difficulty": "0x400",
    "mixhash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "coinbase": "0x3333333333333333333333333333333333333333",
    "alloc": {},
    "config": {"chainId": 987, "homesteadBlock": 0, "eip155Block": 0, "eip158Block": 0,
    "byzantiumBlock": 0}
}
```


Figure 4.17: Genesis block

Explanation on the config file;

- chainId - A unique identifier of the new private blockchain
- homesteadBlock - Homestead is the first production release of Ethereum and since the developers are already using this version the value of this parameter can be left as '0'.
- eip155Block/eip158Block - EIP stands for "Ethereum Improvement Proposals", these were implemented to release Homestead. In a private blockchain

development hard forks aren't needed, hence the parameter value should be left as '0'.

- difficulty - Controls the complexity of the mining puzzle and a lower value enables quicker mining.
- gasLimit - Establishes an upper limit for executing smart contracts.
- alloc - Allows allocation of Ether to a specific address.

The difficulty of block mining is set to a much lower value than in the public Ethereum main network to save computational power. The parameters of the genesis file can be customized according to the requirement. To create the private network, we need to specify the network id parameter in the executing command. This marks the identity of our Ethereum network. It can be replaced with a random number except '1' to create our own network and to prevent others from inadvertently connecting to our network. The Main Ethereum network has a networkid = 1. Then, we created accounts to manipulate our blockchain network. More peers/nodes can be added to our private blockchain by adding nodes' *enode* URL.

4.3 Smart Contracts

The implementation of the smart contract was done in Solidity language which is the programming language to code the smart contracts on Ethereum blockchain. One of the main reasons we chose the Ethereum blockchain implement the proof of concept solution is this feature of ability to write smart contracts embedding contractual clauses using solidity language. Smart contracts eliminate the need of trusted third party to coordinate the transactions between primary and secondary users by a self-executing piece of code coded and deployed in the blockchain network.

4.3.1 Spectral Token

We implemented the SpectatralToken smart contract in order to emphasize the necessary characteristics and behaviours of a digital token which resides a particular frequency band. The following describes these characteristics and behaviours and the way they are implemented in our system.

```
//struct for the spec token
struct SpecToken {
     uint256 _tokenId;
     uint256 _startFreq;
     uint256 _endFreq; }
```

Figure 4.18: Structure of Spectral Token

Figure 4.18 denotes the structure of the spectral token. The spectral token should consist of the token id to uniquely identify each token, the star frequency and the end frequency to derive the frequency band and the location which the particular frequency band belongs to in order to differentiate between the created spectral tokens. The primary user is also included in the structure of the spectral token because it is necessary to identify the actual owner of the spectral token. An event is emitted when the ownership of the token is transferred. This helps the authority to get to know about the fraudulent users of the spectrum. No one can change the ownership of the spectrum without being noticed because of this.

```
mapping(uint256 => SpecToken) public tokenInfos;
      //Array with all tokenIds
mapping(bytes32 => SpecToken[]) public tokenLocation;
      //To filter tokens from location
mapping(address => SpecToken[]) public ownerTokens;
      //To get the tokens of a particular primary user
```

Figure 4.19: The relevant mappings

It is important to retrieve information about the tokens created by the authority. Stakeholders of the spectrum such as primary user, secondary users and the authority needs to get information about the spectrum in different instances. So, the mappings in Figure 4.19 are stored in the blockchain in order to increase the speed of retrieving this information.

```
//create token function
function create(uint256 _startFreq, uint256 _endFreq,
            bytes32 _location, address _primaryUser) public {
      require (owner == msg.sender)
      require (notAvailable(_startFreq, _endFreq, _location),
                  "Token already available. Cannot create Token");
      for (uint256 i = _startFreq; i < _endFreq; i = i + 22) {
            uint256 tokenId = allTokens.length + 1;
            tokenInfos[tokenId]._tokenId = tokenId;
            tokenInfos[tokenId]._startFreq = i;
            tokenInfos[tokenId]._endFreq = i + 22;
            tokenInfos[tokenId]._primaryUser = _primaryUser;
            tokenInfos[tokenId]._location = _location;
            ownerTokens[_primaryUser].push(tokenInfos[tokenId]);
            allTokenInfos.push(tokenInfos[tokenId]);
            tokenLocation[_location].push(tokenInfos[tokenId]);
            tokenByLocBand[_location][_startFreq] = tokenId;
            //Add token to the token list of the primary user
            _mint(_newOwner, tokenId);
            emit token_history(msg.sender, _primaryUser, tokenId,
                  tokenInfos[tokenId]._location,
                  tokenInfos[tokenId]._startFreq, block.timestamp);}
}
```

Figure 4.20: The create token method

Figure 4.20 is the method for the creation of the spectral token. The spectral token should be created only by the authority. In the proposed system, the user which deploys the ITO contract which will be discussed in the next section is considered as the authority. So, before creating the token, it is mandatory to check whether the user asking to create the token is actually the owner of the ITO contract. This is being tested using the first require method. The second require prevents the authority form creating spectral token for the same frequency band and the location for the second time. Once these are tested, the necessary parameters of the spectrum are being set and pushed to the respective arrays in the smart contract.

```
//Function to get token information of the spectral tokens of a
particular frequency band and location

function getTokenByLocBand(bytes32 _location, uint256 _startFreq)

            public returns (address[] owner, uint256 id){

    uint256 tokenId = tokenByLocBand[_location][_startFreq];
```

Figure 4.21: The information retrieval given location and bandwidth

Using the mapping discussed above, figure 4.21 gets the information of the token of a particular frequency band and location. This is important to the authority to get to know about the status of it by comparing the primary user and the current user of that particular spectral token.

Figure 4.22 shows the method which helps the authority to check whether the particular spectral token with the given data has already being sold to a primary user. If it is not sold, the authority can allocate it to the requested primary user or else the request is declined.

The method in figure 4.23 is used to retrieve the current spectral wallet of a particular user.

```
//Check the availability of the token
function notAvailable (uint256 _startFreq, uint256 _endFreq,
            bytes32 _location) public returns (bool){
    bool flag = false;
    for (uint256 i = _startFreq; i < _endFreq; i = i + 22){
        if (tokenByLocBand[_location][_startFreq]==0){
            flag =true;
        }
    }
    return flag;
}
```

Figure 4.22: The function to check the availability of the token

```
//Returns the current owner (the one who can transmit) of th token
function getOwner(uint256 _tokenId) public returns(address owner){
            return ownerOf(_tokenId);
}
```

Figure 4.23: Function to retrieve the token wallet

Figure 4.24 helps the primary user to get to know the information about the tokens which are owned by his/her. Using the methods in figure 4.23 and figure 4.24 we are able to retrieve the advertisable tokens by a particular user. To be a channel advertisable, he/she should be the primary user of that channel as well as at the moment of advertising he/she should be the owner of it.

```
//get all the transmittable tokens of a particular user
function getTokenSet(address _owner) public returns(bytes32[]){
      uint256[] tokens = ownedTokens[_owner];
      uint256 arrLen = tokens.length;
      bytes32[] memory info = new bytes32[] (arrLen * 3);
      uint256 count = 0 ;
      for (uint256 i = 0; i < arrLen ; i ++){
        info[count] = bytes32(tokenInfos[tokens[i]]._startFreq);
        info[count + 1] = bytes32(tokenInfos[tokens[i]]._endFreq);
        info[count + 2] = tokenInfos[tokens[i]]._location;
        count = count + 3;
      }
      return info;
}
```

Figure 4.24: Function to retrieve the transmittable token set of a particular user

The method in Figure 4.25 helps to change the ownership of a spectral token which helps the primary user to lease (change the ownership of the spectral token) it. Before changing the ownership, it is necessary to check whether the receiver of the spectral token and the transferrer of it are valid. If they are not valid the method won't execute. If it is valid, the ownership transferring (the spectral token is removed from the transferrer's Figure 4.18 denotes the structure of the spectral token. The spectral token should consist of the token id to uniquely identify each token, the star frequency and the end frequency to derive the frequency band and the location which the particular frequency band belongs to in order to differentiate between the created spectral tokens. The primary user is also included in the structure of the spectral token because it is necessary to identify the actual owner of the spectral token. An event is emitted when the ownership of the token is transferred. This helps the authority to get to know about the fraudulent users of the spectrum. No one can change the ownership of the spectrum without being noticed

because of this. wallet and added to the receiver's wallet) happen while emitting the *token_history* event.

```
//changing the ownership of a token
function changeOwner(address _oldOwner, address _newOwner,
            uint256 _tokenId) public view returns(address){
      require((ownerOf(_tokenId)== tokenInfos[_tokenId]._primaryUser
            and tokenInfos[_tokenId]._primaryUser == _oldOwner ) or
            (ownerOf(_tokenId)!=tokenInfos[_tokenId]._primaryUser
            and tokenInfos[_tokenId]._primaryUser == _newOwner ))
      transferFrom(_oldOwner,_newOwner,_tokenId);
      emit token_history(_oldOwner, _newOwner, _tokenId,
            tokenInfos[_tokenId]._location,
            tokenInfos[_tokenId]._startFreq, block.timestamp);
      return ownerOf(_tokenId);
}
```

Figure 4.25: Change ownership method

These are the basic functionalities of the Spectral Token contract. Apart from the above-mentioned functionalities, there are others associated with it.

4.3.2 Initial Coin Offering

```
struct SpectrumRequestor{
    uint256 startFreq;
    uint256 endFreq;
    bytes32 location;
    address primaryUser;
}
```

Figure 4.26: Structure of the spectrum requestor

Figure 4.26 shows the structure of the spectrum requestor. Once a primary request for a particular frequency band using the method mentioned in Figure 4.27, the information is set and stored in the ITO contract until the authority accepts or declines the request.

```
function pushRequest(bytes32 _location, uint256 _startFreq,
                        uint256 _endFreq) public {
      SpectrumRequestor memory spec;
      spec.primaryUser = msg.sender;
      spec.startFreq = _startFreq;
      spec.endFreq = _endFreq;
      spec.location = _location;
      spectrumInfoArray.push(spec);
}
```

Figure 4.27: Request for frequency band

If the requested primary user has the necessary qualifications, the authority is accepting the request while creating the respective spectral token (Figure 4.28) and removing the request using the method in Figure 4.29. If the primary does not meet the necessary standards, the authority can decline the request.

```
function createToken(uint256 requestId, uint256 _startFreq,
            uint256 _endFreq, bytes32 _location, address _newOwner,
            address addr) public {
      SpectralToken spec = SpectralToken(addr);
      spec.create(_startFreq,  _endFreq,  _location,  _newOwner);
      declineRequest(requestId);
}
```

Figure 4.28: Create token method

```
function declineRequest(uint256 requestId) public returns(bool){
      require (requestId < spectrumInfoArray.length);
      for(uint256 i=requestId; i < spectrumInfoArray.length-1; i++){
            spectrumInfoArray[i] = spectrumInfoArray[i+1];
      }
      delete spectrumInfoArray[spectrumInfoArray.length-1];
      spectrumInfoArray.length--;
      return true;
}
```

Figure 4.29: Decline of the request

4.3.3 Advertising smart contract

We implemented the advertising smart contract so that it is customizable to have two different leasing policies called competitive bidding and first come first served bidding. The primary user can customize this leasing policy when he/she is advertising the frequency band. They also can advertise the frequency band to lease the channel at the time of advertising or for a future time frame. If the frequency band is advertised for now primary user can specify a percentage from the leasing time as the bidding time. So, there are 4 different ways of customizing the advertising facilitated in our platform. As these advertising smart contracts are customizable primary user have to deploy a contract per advertising using the smart contract template provided by the platform. Below show are some important methods from the implementation of the smart contract.

Figure 4.30 shows the advertise contract creation method for which successful execution would deploy a smart contract on the blockchain. It checks whether the contract is in the *initiated* state which we discussed in the system design. The platform enforces that the contract deployment can only be done by a primary user of a frequency band. Therefore, one of the main concerns of secondary users, whether the advertised spectral token is owned by a primary user is solved. Platform uses another master contract which is

triggered on deployment every advertising smart contract. So that we can check whether there is ongoing advertisement for the same spectral token. If so, the deployment would be unsuccessful. And also, secondary users should be able to retrieve the details of the advertised smart contracts when search by the location. They can do so by querying the events triggered by master contracts master contract.

```
mapping(address => Bidder) public bidder_info;
constructor (address _token_addr, address _advertised_addr, bytes32
_location, uint _min_bid, uint        _max_bid, uint _lend_start,
uint _lend_exp, uint _bid_start, uint _bid_exp, uint _start_freq,
uint _end_freq, uint _token, bool _fcfs, bool _future) public {
       token_contract = SpectralToken(_token_addr);
       contracts = AdvertisedContracts(_advertised_addr);
       //check whether the contract owner owns the tokens
       require(token_contract.getPrimaryUser(token) == msg.sender,
"Advertising invalid token");
       //check whether there is an ongoing advertisement for the
channel
       require(contracts.check_advertised(_token, _lend_start),
"Token already advertised");
       require (_lend_start < _lend_exp, "Non exisiting leasing
time");
       require(block.timestamp < _lend_exp, "No future lending time
specified");
       require(_end_freq > _start_freq, "Not enough bandwidth
provided");
       require(_min_bid >= 0, "Minimum bid not specified");
       require (_bid_start < _bid_exp, "Non exisiting bidding
time");
       require (_bid_exp <= _lend_start, "bidding time overlaps
leasing time");
       require(block.timestamp < _bid_exp, "No future bidding time
specified");
       primary_user = msg.sender; max_bid = _max_bid; bid_start =
_bid_start; bid_exp = _bid_exp;
       primary_user = msg.sender; min_bid = _min_bid; lend_start =
_lend_start; lend_exp = _lend_exp;
       start_freq = _start_freq; fcfs = _fcfs; future = _future;
end_freq = _end_freq; location = _location;
       token = _token; lended = false; leased = false; no_of_bidders
= 0; best_bid = 0; }
```

Figure 4.30: Advertise contract creation method

Figure 4.31 shows bid method from the implemented advertise smart contract. Secondary users can bid for the frequency band calling this bid method. Bidders can only bid for the frequency band only when the corresponding smart contract is in the bidding state. Therefore, the necessary conditions are checked to enforce those rules. As this method is a payable method, bidding amount is transferred from the bidder's ether wallet to the smart contract. This bid method handles the leasing policies appropriately and update the state of the smart contract accordingly. Each contract keeps a map to store the bidders for the frequency channel so that it can select the best bidder when secondary users invoke this bid method.

```
function bid() public payable returns(bool){
      require(!lended, "already lended");
      require(msg.sender != primary_user, "primary user can't bid");
      require(block.timestamp >= bid_start, "Bidding time not
                                          started yet");
      require(block.timestamp <= bid_exp, "Bidding time is over");
      require(msg.value >= min_bid, "Bid is not enough");
      if(fcfs){
           best_bid = msg.value;  best_bidder = msg.sender;
           no_of_bidders ++;lended = true;
           if (!future){
               emit token_transfer(token, primary_user, msg.sender);
               leased = true; }
           return true; }
      else{
           require(!check_bidder_exists(msg.sender));
           require(msg.value > best_bid, "Higher bid exists");
           bidder_info[msg.sender].bid = msg.value;
           if (max_bid > 0 && msg.value >= max_bid){
               best_bid = msg.value; best_bidder = msg.sender;
               lended = true; }
           else if (best_bid < msg.value){
               best_bid = msg.value; best_bidder = msg.sender; }
           no_of_bidders++;  bidders.push(msg.sender);
           return true; }
}
```

Figure 4.31: Bid method of advertise contract

```
function lease_channel() public returns(bool){
       require(msg.sender == primary_user,
                  "No authority to transfer token");
       require(!leased, "already leased");
       require(block.timestamp >= lend_start,
                  "leasing start time not met");
       require(block.timestamp < lend_exp,
                  "leasing time has expired");
       require(best_bid > 0, "No bidders bidded for the channel");
       transfer_token(msg.sender, best_bidder, token);
       lended = true;
       leased = true;
       for(uint i = 0; i < bidders.length; i++){
           if(bidders[i] != best_bidder){
               bidders[i].transfer(bidder_info[bidders[i]].bid);
               bidder_info[bidders[i]].bid = 0; }}
       return true; }

function lease_channel() public returns(bool){
       require(msg.sender == primary_user,
                  "No authority to transfer token");
       require(!leased, "already leased");
       require(block.timestamp >= lend_start,
                  "leasing start time not met");
       require(block.timestamp < lend_exp,
                  "leasing time has expired");
       require(best_bid > 0, "No bidders bidded for the channel");
       transfer_token(best_bidder, msg.sender, token);
       lended = true;
       leased = true;
       for(uint i = 0; i < bidders.length; i++){
           if(bidders[i] != best_bidder){
               bidders[i].transfer(bidder_info[bidders[i]].bid);
               bidder_info[bidders[i]].bid = 0;}}
       return true; }

function transfer_token(address _from, address _to, uint _token_id)
                        private returns(bool){
       token_contract.changeOwner(_from, _to, _token_id);
       return true; }
```

Figure 4.32: Transfer ownership and transfer assets of advertising contract

Figure 4.32 shows the implementation of the smart contract for transferring the ownership of the spectral token from primary user to the secondary user and transferring assets back

to the primary user. Transferring ownership of the spectral token from primary user to the secondary user can be done only when the smart contract is in the *lended* state. This method can only be called by the primary user of the token. After the leasing time is over the primary user can transfer the token back while transferring the best bid to his wallet.

4.3.4 Requesting smart contract

The request contract is deployed at the deployment of the system. As request can be done only on real time one contract deployed on the system can handle the requests. The request contract keeps a mapping from token id to requestor addresses who are the secondary users for who are willing to use the vacant frequency band. Requesters have to specify the payment they are willing to pay, expiration time of the request as well as the time duration they want to lease the channel in the request. Those requests are stored in the smart contract and the payments are also transferred to the smart contract on the time of request as a bid. Figure 4.33 shows the request method of the smart contract.

```
function request_token(bytes32 _location, uint _start_freq,
      uint _end_freq, uint _duration, uint _exp_time, uint _token,
      address _primary_user) public payable returns(bool){
      require(block.timestamp < _exp_time, "expiration time is
                  smaller than the current time");
      require(token_contract.getOwner(_token) == _primary_user,
                  "token already leased");
      require(block.timestamp > grants[_token].exp_time, "token
                  already leased");
      requests[_token][msg.sender].primary_user = _primary_user;
      requests[_token][msg.sender].exp_time = _exp_time;
      requests[_token][msg.sender].duration = _duration;
      requests[_token][msg.sender].bid = msg.value;
      requests[_token][msg.sender].start_freq = _start_freq;
      requests[_token][msg.sender].end_freq = _end_freq;
      requests[_token][msg.sender].location = _location;
      requests[_token][msg.sender].declined = false;
      requests[_token][msg.sender].granted = false;
      requestors[_token].push(msg.sender);
      emit request(_token, _location, _primary_user, msg.value,
                  _exp_time, _start_freq, _end_freq);
      return true; }
```

Figure 4.33: Request method

When the request is made an event is triggered in the blockchain. So, the primary users who are connected the blockchain listening to the events triggered from the request contract are notified. So, the primary user of the requested frequency channel can either accept or decline the requests according to the channel availability. If multiple secondary users have requested the same channel, in close time durations the accept method of the smart contract would select the best bidder and transfer the ownership of the spectral token to the requestor while transferring the best bid to the primary user. Figure 4.34 shows the ***accept*** method of the request smart contract. If the frequency band is already leased to another secondary user either by accepting a request or by advertising, the current owner of the spectral token is not the primary user. Therefore, the accept method can be executed if the spectral token is not leased. The secondary users are notified about the grant of the spectral token by triggering an event on the blockchain. So secondary users can listen to that event and transmit data on the granted channel for requested time period. On the grant

expiration time primary user can transfer back the token as similar to the advertising model. However, ether transferrin is not taking place in the transfer back method as ether is transferred to the corresponding primary user at the request granting time.

```
function accept_requests(uint _token, uint _end_time) public
                                    returns(bool){
      address best_bidder;
      uint best_bid = 0;
      require(token_contract.getPrimaryUser(_token) == msg.sender,
                  No Authority to perform the request");
      require(token_contract.getOwner(_token) == msg.sender,
                  "No Authority to perform the request");
      for(uint i = 0; i < requestors[_token].length; i++){
          if(block.timestamp <
                  requests[_token][requestors[_token][i]].exp_time){
              if(block.timestamp +
                  requests[_token][requestors[_token][i]].duration >
                  _end_time){
                  requests[_token][requestors[_token][i]].declined =
                                                          true;}
              else if(requests[_token][requestors[_token][i]].bid >
                  best_bid){
                  best_bid =
                        requests[_token][requestors[_token][i]].bid;
                  best_bidder = requestors[_token][i];}}
           else{
               requests[_token][requestors[_token][i]].declined =
                                                          true;}}
      if(best_bid > 0){
          transfer_token(msg.sender, best_bidder, _token);
          grants[_token].primary_user = msg.sender;
          grants[_token].secondary_user = best_bidder;
          grants[_token].exp_time = block.timestamp +
                  requests[_token][best_bidder].duration;
          requests[_token][best_bidder].bid = 0;
          msg.sender.transfer(best_bid);
          emit token_transfer(_token, msg.sender, best_bidder);
          return true;}
      else{ return false; }
}
```

Figure 4.34: Accept method of request contract

5 PERFORMANCE EVALUATION AND DISCUSSION

5.1 Proof of Concept Design

In our PoC we developed a platform using 2.4 GHz and 5 GHz ISM band assuming that the authority node can issue licenses for each frequency band in 2.4 GHz and 5 GHz ISM band. Our experimental setup composed of six laptops (five as mining nodes) and six smartphones. Here, mobile phones act as transmitters, spectrum sensing devices, and user interface providers, while laptops act as blockchain nodes and back-end for mobile devices. The communication between the blockchain and devices is carried out via channel 14 (2473-2495 MHz). Five of the laptops had DualCore Intel processors (Core i5-7200U x 2, i7-4510U x 2, and i7-7500U) while other was a Quad-Core Intel Core i7-8550U processor. All laptops had 8GB of RAM.

5.2 Performance Analysis

The performance analysis of the system was done respect to the time and throughput because time and throughput are critical evaluation parameters of the system as it deals with white spaces that appear and disappear in even time of sub seconds. We evaluated the performance of the PoC platform by emulating a set of advertising-based DSA transactions using FCFS leasing policy. Competitive bidding model is not considered in performance analysis because smart contracts will remain pending until the end of bidding time. Here advertising includes setting necessary parameters for leasing and deploying the smart contract to the blockchain. Leasing those frequency bands include transferring the spectral token between primary user and secondary user. As seen in Figure 5.1 and 5.2 we measure latency and throughput while varying the number of concurrent users. Moreover, Figure 5.3 shows that the latency of 50 concurrent transactions against varying number of miners.

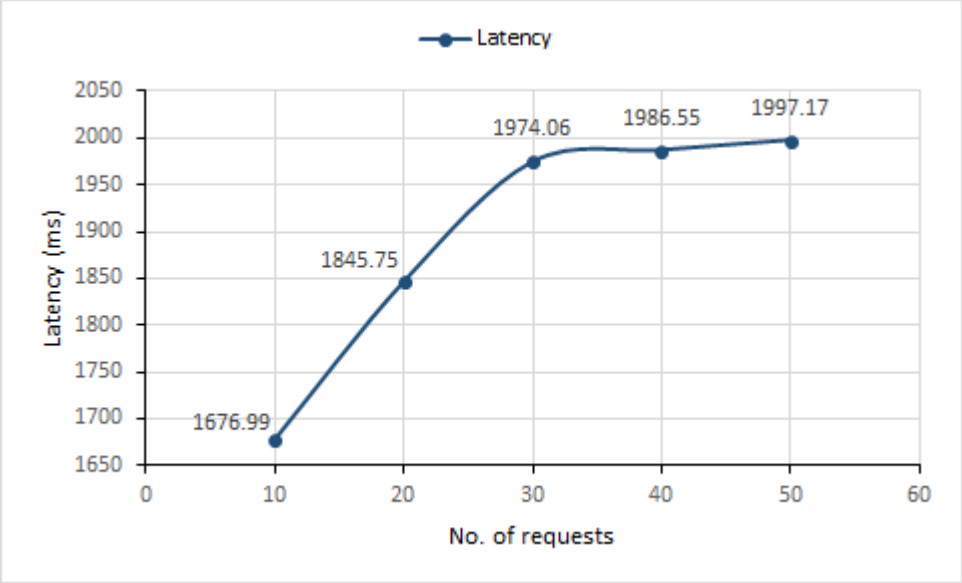


Figure 5.1: Latency of FCFS leasing transactions

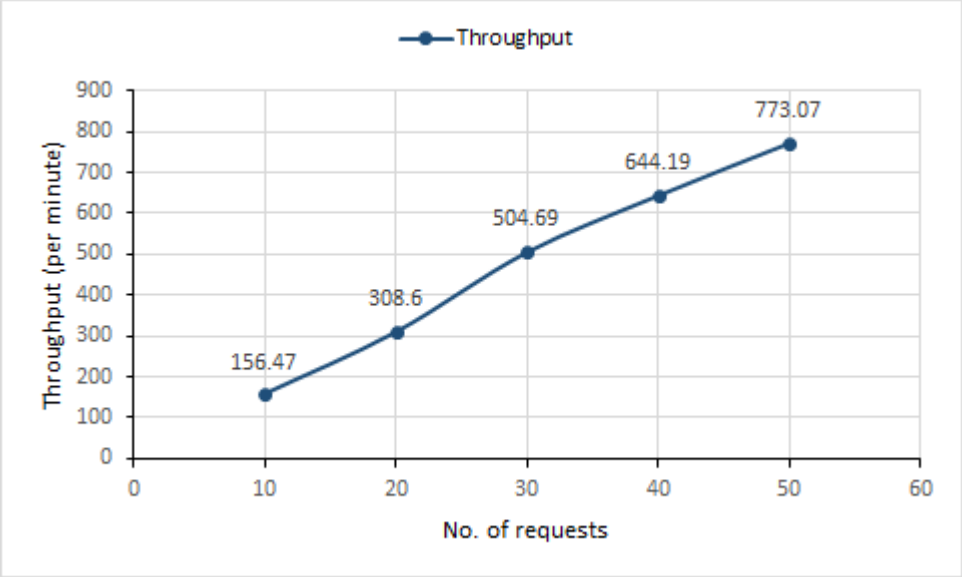


Figure 5.2: Throughput of FCFS leasing transactions.

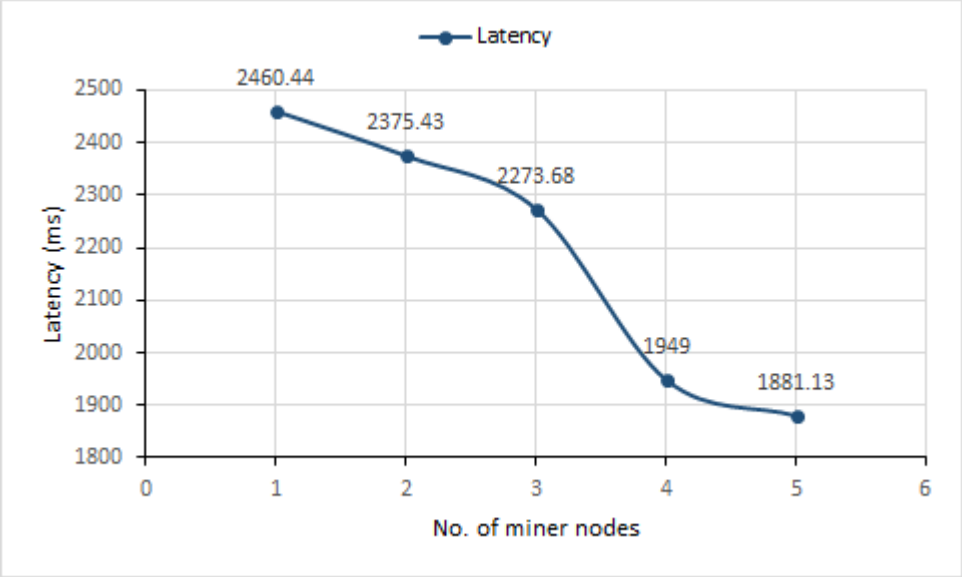


Figure 5.3: Transaction latency vs. number of miners (50 concurrent transactions)

As shown in graph in Figure 5.1 the number of concurrent users increases the latency gets saturated. Latency is the delay from input into a system to desired outcome. So, performance of the system is directly affected by the latency. If the latency tends to increase with the number of users, then the system response time is highly degrading with the increase of the transactions processing in the system. But as observed in the testing the latency tends to saturate which is a good performance characteristic of the system. Because the latency tends to saturate with the number of concurrent users the throughput of the system tends to increase as the transactions handled by the system per minute increases. As shown in graph in Figure 5.3 the latency of transactions tends to decrease as the number of miners increase. This is due to the increase in overall computing power of blockchain miners which results in a relatively fast generation of new blocks by distributing computational power needed to confirm transactions among miners. Here we can see that the throughput and latency are limited by the block generation process than the execution of smart contracts. It can be seen that even with modest hardware more than 10 transactions per second is achievable. This throughput is likely to be sufficient if transactions are for relatively longer spectrum sharing which we can conclude as minutes range.

5.3 Maintaining integrity of transactions

Assume that a fraudulent primary user P_f owns a spectral token T_1 and leases it to a secondary user S_1 through smart contract C_1. However, if P_f tries to lease T_1 to another secondary user S_2 (perhaps to gain a higher fee) using a different smart contract C_2, the request will fail. This is because blockchain miners will detect that T_1 required by C_2 is already in the leased state and no longer an asset of P_f, as it is temporarily transferred to S_1. While P_f may try to create a fork of the blockchain by mining another version of the same block, future blocks build by the majority of honest miners will eventually replace the block with the fraudulent transaction, as per the longest chain wins consensus mechanism proposed by Nakamoto [21]. Therefore, the transaction between P_f and S_2 will never take place.

Moreover, a secondary user cannot avoid paying for the transferred frequency bands, as a spectral token is transferred only when the cryptocurrency is charged from the secondary user's account by the smart contract at the bidding time as discussed. Thus, the proposed advertising and sensing-based DSA smart contracts ensure that both the primary users and secondary users behave as expected.

6 CONCLUSION

6.1 Summary

In this work, we proposed a token based dynamic spectrum sharing platform using blockchain and smart contracts which enhances the efficient use of the spectrum by allowing its primary users to share the spectrum access with secondary users. Here, blockchain is used as a distributed system which helps to maintain the integrity and transparency of the spectral transactions. This system outperforms the traditional spectrum sharing mechanisms in terms of time because it supports features such as verifying the ownership of the frequency band, transferring the license to a secondary user without the mediation of a trusted third party. Here, the utility and performance of the proposed system is evaluated using the implementation done in ISM band.

The proposed system consists of two sub-models namely, advertising and sensing, according to the initiation method of the spectrum sharing process. A smart contract is initiated at the beginning of each of these scenarios. The primary user gets the permission to decide the criteria to select a lessee of a particular channel. This can either be done by competitive bidding or FCFS mechanisms. Once the lessee is selected, the asset transferring (Spectral Token and Ether) between primary user and secondary user takes place and the secondary user receives license to transmit in the particular frequency band. Ultimately when the time duration of the contract expires, the Spectral Token which depicts the ownership (license) of the frequency band is transferred back to the primary user through the smart contract. Hence, ending the smart contract. All these transactions are validated and recorded in the Blockchain by miners.

The use of blockchain and smart contract in the system enhances the trust on the system and ensures both the primary users and secondary users share the spectrum as per regulations coded into the platform, enforcing the sequential access to the system and the licensed usage of the spectrum. Moreover, it enables license transfer and verification without the mediation of a trusted third party.

We demonstrate utility and performance of the proposed platform by developing it using the Ethereum blockchain and a set of wireless devices. Due to the relatively high latency introduced by the blockchain mining process, the proposed platform is desirable for DSA in CBRS, IEEE 802.22 WRAN, and Small-Cell as a Service use cases where spectrum sharing is relatively long lived and into the future. The results of the PoC showed that the latency of the transaction reduced with the increasing number of miners. Also, the latency of the responses increased in the beginning and then stabilized with increasement of the number of requests (transactions). The throughput of the system increased linearly with increasing number of requests (transactions). These performances conclude that the system proposed is at a relatively high standard.

6.2 Limitations

The price of the spectrum should change with the time and demand. In this system we have not change the price of the spectrum with the time. Further, in our system, the minimum bid amount is fixed. That is, the primary user will not lease the channel below a certain amount. This may lead to not lease the frequency band even for a single secondary user, hence reducing the profitability of the primary user. In order to deal with highly-dynamic cognitive radio systems, the speed of the current system is insufficient.

6.3 Future Work

The varying price of the spectrum with the time and demand can be overcome in future, by providing a cost function to the frequency band which varies with time and can be decided by the primary user. Further, in our system, the minimum bid amount is fixed. That is, the primary user will not lease the channel below a certain amount. But we can introduce a negotiation model where, when the secondary user tries to bid for a lower price, after a negotiation phase, the primary and secondary could agree upon a value (which is between the secondary's initial bid amount and the primary's initial minimum bid). This limitation of our system can be implemented in the future versions of the system.

In order to deal with highly-dynamic cognitive radio systems, the speed of the current system is insufficient. So, we need to use a highly efficient and speed blockchain network in order to achieve this. The bottleneck of speed which occurs due to the blockchain can only be avoided by using an efficient and fast blockchain network. Upgrading the system with the advancement of blockchain technology will help to get rid of this bottleneck with the time. Therefore, these improvements will help to improve the quality and the generality of the platform.

References

[1] Z. Ji and K. J. R. Liu, “Cognitive radios for dynamic spectrum access - dynamic spectrum sharing: A game theoretical overview,” *IEEE Communications Magazine*, vol. 45, no. 5, pp. 88–94, May 2007.

[2] J. Jeon *et al*., “LTE in the unlicensed spectrum: Evaluating coexistence mechanisms,” in *IEEE Globecom Workshops (GC Wkshps ’14)*, Dec 2014, pp. 740–745.

[3] J. Andrews *et al*., “What will 5G be?” *IEEE J. Sel. Areas Commun*., vol. 32, no. 6, pp. 1065–1082, June 2014.

[4] V. Valenta *et al*., “Survey on spectrum utilization in europe: Measurements, analyses and observations,” in *5th Intl. Conf. on Cognitive Radio Oriented Wireless Networks and Communications*, June 2010.

[5] H. Griffiths *et al*., “Challenge problems in spectrum engineering and waveform diversity,” in *IEEE Radar Conf. (RadarCon ’13)*, April 2013, pp. 1–5.

[6] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13– 18, Aug 1999.

[7] M. Crosby *et al*., “Blockchain technology: Beyond bitcoin,” *Applied Innovation Review*, vol. 2, pp. 6–11, 2016.

[8] “Solidity,” [Accessed: 12-Oct-2018]. [Online]. Available: https://solidity. readthedocs.io/en/v0.4.25/

[9] C. Cachin, "Architecture of the Hyperledger blockchain fabric", *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[10] S. Yrjola, “Analysis of blockchain use cases in the citizens broadband radio service spectrum sharing concept,” in *Intl. Conf. on Cognitive Radio Oriented Wireless Networks*, 09 2017, pp. 128–139.

[11] C. Stevenson *et al*., “IEEE 802.22: The first cognitive radio wireless regional area network standard,” *IEEE Communications Magazine*, vol. 47, no. 1, pp. 130–138, January 2009.

[12] S. Bayhan, A. Zubow, and A. Wolisz, “Spass: Spectrum sensing as a service via smart contracts,” in *IEEE Intl. Symp. on Dynamic Spectrum Access Networks (DySPAN ’18)*, Aug 2018.

[13] K. Kotobi and S. G. Bilen, “Blockchain-enabled spectrum access in cognitive radio networks,” in *Wireless Telecommunications Symp. (WTS ’17)*, April 2017, pp. 1–6.

[14] E. D. Pascale *et al*., “Smart contract SLAs for dense small-cell-as-a-service,” in *arXiv preprint arXiv:1703.04502*, Mar 2017.

[15] V. Buterin, “A next-generation smart contract and decentralized application platform,” 2014.

[16] S. Jilani, "Spectrum Allocation Methods: Studying Allocation through Auctions", *Journal of Economics, Business and Management*, vol. 3, no. 7, pp. 742-745, 2015.

[17] I. Akyildiz, W. Lee, M. Vuran and S. Mohanty, "A survey on spectrum management in cognitive radio networks", *IEEE Communications Magazine*, vol. 46, no. 4, pp. 40-48, 2008.

[18] Y. Liang, K. Chen, G. Li and P. Mahonen, "Cognitive radio networking and communications: an overview", *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386-3407, 2011.

[19] M. Song *et al.*, "Dynamic Spectrum Access: From Cognitive Radio To Network Radio", in *IEEE*, 2012, pp. 23-29.

[20] Z. Luo and S. Zhang, "Dynamic Spectrum Management: Complexity and Duality", *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57-73, 2008.

[21] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[22] Z. Kakushadze and R. Russo, Jr., "Blockchain: Data Malls, Coin Economies and Keyless Payments", *SSRN Electronic Journal*, 2018.

[23] A. Sehra, P. Smith and P. Gomes, "Economics of Initial Coin Offerings", 2017.

[24] “ERC20 Token Standard,” [Accessed: 12-Oct-2018]. [Online]. Available: https://theethereum.wiki/w/index.php/ERC20 Token Standard

[25] “ERC721 Token,” [Accessed: 12-Oct-2018]. [Online]. Available: https://moac-docs.readthedocs.io/en/latest/ERC721.html

[26] “Electronic Code of Federal Regulations,” [Accessed: 12-Oct-2018]. [Online]. Available: https://www.ecfr.gov/cgi-bin/text-idx?node=pt47.5. 96&rgn=div5

[27] "The Product Manager’s guide to the Blockchain — Part 1", *Hacker Noon*, 2018. [Online]. Available: https://hackernoon.com/the-product-managers-guide-to-the-blockchain-part-1-fb95dfb7af31. [Accessed: 18- Nov- 2018].