IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 September 2023, accepted 31 October 2023, date of publication 30 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337880

RESEARCH ARTICLE

Streamlining 5G Spectrum Leasing

MUKESH THAKUR[1], YKI KORTESNIEMI[2], AND DMITRIJ LAGUTIN[2]
[1]Ericsson, 02420 Kirkkonummi, Finland
[2]Department of Information and Communications Engineering, School of Electrical Engineering, Aalto University, 02150 Espoo, Finland
Corresponding author: Mukesh Thakur (mukesh.thakur@ericsson.com)

ABSTRACT The growing demand for wireless communications has driven the development of new networking technologies; however, it also makes efficient usage of the limited frequency spectrum increasingly important. Leasing an unused spectrum on a short-term basis would help increase utilization, but complicated lease negotiations usually make it financially unviable. To address this, the entire leasing process, including negotiations, needs to be streamlined. This paper presents an automated spectrum marketplace for leasing that allows frequency bands to be placed for anonymous auctions between qualified (sub)leasers. The solution utilizes decentralized identifiers and verifiable credentials for entity privacy, and distributed ledgers for the automation and auditing of contract agreements. The prototype implementation demonstrates that the solution significantly streamlines the spectrum leasing process compared with the current solutions.

INDEX TERMS Frequency auction, privacy, automation, accountability, distributed ledgers, decentralized identifiers, verifiable credentials.

I. INTRODUCTION

With the Internet of Things (IoT) and increasing automation and data exchange in industries, the demand for wireless broadband has been increasing exponentially [1], putting increasing pressure on the limited radio spectrum [2].

To tackle spectrum scarcity, *Spectrum Regulators* have devised varying regulations that allow a spectrum owner/lessor, such as an operator to temporarily lease a part of their unused spectrum for a defined duration and within a certain spectrum coverage area to an authorized party [1]. Currently, there are two popular approaches for spectrum leasing. In the US, a spectrum management system called the Universal Licensing System (ULS) [3] is, administered by the government organization Federal Communications Commission [4]. While, in Europe, the spectrum management framework known as evolved Licensed Shared Access (eLSA) [1] has been introduced. Both are based on a centralized system and rely on a manual process of negotiation that is inefficient and, time-consuming, typically lacks privacy protection for the participant, and requires a third-party to audit the transactions [5]. Streamlining the spectrum leasing agreement process could potentially significantly reduce the related costs and make the spectrum more dynamically available for short-term leases, thus opening up new business opportunities and further enhancing spectrum usage.

To streamline the spectrum leasing agreement, negotiation, and auditing should be achieved swiftly and with minimal manual involvement. Moreover, the revelation of the exact spectrum bands being shared carries the risk of exposing sensitive business secrets and, potentially disclosing the proprietary technologies under evaluation. Similarly, revealing pricing details or participants identities during negotiations could e.g., trigger bidding among rival companies, thus increasing prices. Hence, certain parameters within the agreement must be reserved for only chosen members, and ensuring that participants' anonymity (at least during the negotiation) is important. In addition, in countries such as Germany, there are regulations that mandate that participants must be able to retain their anonymity even after reaching an agreement, further necessitating the anonymity of the participants [6].

Thus, there is a need for an alternative solution to streamline spectrum leasing. The key research questions in designing such a solution include the following:

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-Hoon Kim.

1) How can the current highly manual spectrum leasing process be automated to increase efficiency and reduce complexity and time-consumption?
2) How can the privacy of the participants be protected during the spectrum leasing process?
3) How can the auditability of spectrum leasing transactions be ensured in a transparent and verifiable manner?

This paper presents the design and prototype of an *Automated Spectrum Marketplace (ASM)* that utilizes blockchain technology and Self-Sovereign Identities (SSIs) to streamline the leasing process. The prototype is implemented using Hyperledger Fabric [7] as the underlying blockchain platform and SSI based on the Horizon 2020 Privacy-Preserving Self-Sovereign Identities - IoT-NGIN framework [8]. The results show that ASM significantly reduces (from weeks or months to minutes) the latency of the leasing process compared to current solutions. Additionally, the use of blockchain and SSI technologies in the ASM prototype ensures the privacy of participants and transactions.

The rest of the paper is structured as follows: Section II describes the need for spectrum leasing, reviews current spectrum leasing solutions and their key limitations, and identifies the requirements for an automated spectrum leasing solution. Section III introduces the main technologies used to build an automated spectrum leasing solution. Section IV reviews related work in the field. The design of the automated spectrum leasing solution is presented in Section V and the prototype is described in Section VI. The measurements of the prototype are presented in Section VII, and Section VIII analyzes the design and results of the solution. Section IX discusses the implications, and Section X concludes the paper.

II. SPECTRUM LEASING

Spectrum Leasing is a mechanism that allows spectrum license owners to lease their spectrum bands to qualified users [1] for a certain duration[1] [9]. In this context, spectrum license owners are called *Lessors* and typically include mobile network operators and government agencies in-charge of regulating the spectrum. Spectrum users, on the other hand are called *Lessees* and include organizations such as smart factories, industries, and enterprises, as well as event and shipping companies that need access to spectrum bands for specific (short or longer) time periods.

This section describes a generic spectrum leasing use case and introduces the key actors (Regulators, Mobile Network Operators and Organizations) involved in the leasing process. It also describes the most popular current spectrum leasing solutions including evolved Licensed Shared Access (eLSA) [1] and Universal Licensing System (ULS) [3], and finally, derives the requirements for an automated spectrum leasing solution.

[1]There are also other mechanisms of spectrum access, such as spectrum sharing, spectrum transfer, but this paper only focuses on spectrum leasing.

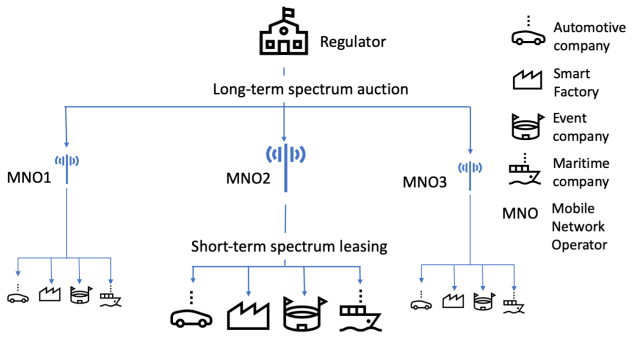

FIGURE 1. Entities in spectrum leasing use case.

A. USE CASES FOR SPECTRUM LEASING

A typical use case has three main types of entities as shown in Figure 1. The *Spectrum Regulator*, hereafter also *Regulator*, is typically a government authority, for example, Traficom [10] in Finland, which is in-charge of enforcing spectrum usage regulations such as defining and monitoring spectrum licenses, auction, and leasing rules [2]. *Mobile Network Operators (MNOs)* are telecommunications service providers that acquire spectrum licenses at auctions from the Regulator to provide wireless connectivity services. Finally, *organizations* such as enterprises/industries/factories occasionally require spectrum leasing to ensure sufficient network capacity at their disposal.

The increasing demand for short-term spectrum access is driven by exponential growth in access to networks. For example, automotive companies are gearing towards the autonomous era, where on-board sensors of cars generate terabytes of data to monitor their status, sense their surroundings, and act on behalf of humans. These data are further uploaded to the cloud for post-processing, generating intelligence, and/or optimizing the routes. Subsequently, the processed information is downloaded for follow-up journeys [11]. To ensure high-speed data transfer and ultra-reliable connectivity, short-term dedicated spectrum access is required, which the vehicle can request for a specific spectrum band [12]. However, to obtain short-term access to a spectrum band, the vehicle company must have a pre-agreed spectrum license agreement with a spectrum owner, for example, MNOs. Furthermore, short-term spectrum access is not the preferred method of spectrum access, because obtaining spectrum license is a lengthy and expensive process, as it is mostly manual and involves multiple hierarchical steps. Thus, obtaining spectrum access for short-term might not be cost-efficient for automotive companies.

Similarly, autonomous and remotely controlled ships are becoming a reality in the maritime industry. These autonomous ships can monitor their own status, use predictive diagnostics, sense the environment, communicate obtained information within and outside of the ship (other ferries, shore stations etc.), define possible actions, and act without human supervision [13], [14]. The most critical component enabling autonomous ships is the connectivity solution, both on-ship

and between ships and remote parties. For connectivity, ships can rely on solutions provided by local operators using either an unlicensed or licensed spectrum. With an unlicensed spectrum, there is no interference protection, and reliable connectivity cannot be guaranteed, whereas with a licensed spectrum reliable connectivity without (purposeful) interference can be guaranteed. However, obtaining spectrum licenses can be difficult for ships with international voyages.

Manufacturing facilities are increasingly using arrays of drones and augmented reality (AR) devices for tasks such as visual inventory and safety inspections [15]. During production or safety checks, the demand for spectrum bandwidth can be substantial, due to the concurrent use of numerous drones and AR devices. In situations, where a sufficient spectrum is lacking, short-term spectrum leases may be a solution. Conversely, automated factories may have a spectrum for their use. However, during the holiday season or periods of low production activity, not all of the spectrum can be utilized. These unused portions could be leased to external parties, which could provide economic benefits to the factories.

Moreover, in recent years, event companies have been aiming to deliver a better fan experience, which includes instant replays of game highlights from various point-of-view perspectives and on-demand live-streaming in ultra-high definition with augmented and virtual reality for a more immersive experience [12]. To achieve fan experience, event companies can require very high upload (on the order of 10 Gbit/s) and download (20 Gbit/s) speeds at low latency (1-100 millisecond) and the capacity to simultaneously connect large numbers of (IoT) devices at the stadium without compromising the connection quality [16], [17]. To provide such connectivity, the event companies require a dedicated, ultra-reliable network for the duration of the event, which can be accomplished with short-term access to a licensed spectrum.

Other examples of the need for short-term spectrum access include shopping malls during sale days (to handle additional customer requests) and hospitals during flu seasons (to support additional patient requests).

Big enterprises, who, for example, own multiple factories, can buy a licensed spectrum for their factories for long-term. However, for small and medium sized companies, issues related to occasional (short-term) spectrum access are twofold: first, it is difficult to *efficiently acquire access to the licensed spectrum* and second, the *privacy of users involved in spectrum leasing is not guaranteed with current solutions*. This is a concern for companies, as the spectrum bands they access may be considered a business secret. For example, a company may be testing proprietary technology, and revealing information about the spectrum band they use could hint at the technology they are testing. Further, rival companies may also compete for the same spectrum bands, driving up the price of spectrum access. Thus, prompt access to the short-term licensed spectrum, anonymity of participants, and selective data disclosure are key elements in this short-term spectrum access.

Leasing may require the lessee to bring their own hardware. However, this is not a major issue as e.g., 5G NR Small Cells [18], which are cheaper than cellular network towers [19], make these requirements viable. Alternatively, lessees could even lease hardware from third-party companies for particularly limited needs.

B. CURRENT SPECTRUM LEASING PRACTICES

The spectrum used to provide wireless communication services is a limited natural resource [20]; therefore, its usage is actively governed. The Regulator often auctions long-term (10+ years) spectrum licenses to the MNOs, who lease the unused spectrum bands to the organizations for varying durations, ranging from months to multiple years [1].

Spectrum Regulators have been investigating two distinct approaches to address the high demand for spectrum access. The first approach is to provide an additional spectrum by e.g., using the higher frequency bands above 6GHz, while the second approach enables efficient use of existing unused spectrum bands [2]. The challenge for the first approach is that most of the available communication devices that use spectrum do not support higher frequency bands; thus, moving to higher frequencies requires an update to existing devices or new devices altogether. The 5G New Radio (NR) [21] is an example of a technology that is expanding to higher frequencies. The challenge for the second approach is that most of the spectrum is fully allocated to spectrum users, who oppose re-purposing their spectrum to other stakeholders [2].

In this context, regulators such as the European Conference of Postal and Telecommunications Administrations (CEPT) [22] in Europe, along with standardization bodies such as the European Telecommunications Standards Institute (ETSI) [23] are collaborating. Similarly, the regulator body, the Federal Communication Commission (FCC) [4] in the US, and specification bodies such as WinnForum [24] and OnGo Alliance [25], are working on specifications to allow efficient access to unused spectrum bands [2].

At a given point, a spectrum owner (lessor) hardly utilizes all the spectrum bands they possess, so these unused bands could be leased for the short-term with interested parties (lessees) via leasing agreements. This agreememts describe the terms and conditions of spectrum leasing including price, duration, location, bands etc [1], [5]. There are two main reasons why lessors may be motivated to share their bands. First, lessors could obtain direct economic benefits by leasing idle, unused bands, and second, regulators in different countries have devised directives and policies to promote spectrum access. For example, the European Union has enacted directives that the member state shall allow the leasing of spectrum in harmonized bands such as 790-862 MHz, 880-915 MHz, 3.4-3.8 GHz, etc. [26]. Based on the EU directive, Regulators in the United Kingdom, Ireland, Sweden, Germany, and Finland have devised policies

for the commercial use of spectrum [27]. Similarly, in the United States, the FCC has devised Citizen Broadband Radio Service (CBRS) [28] regulation for commercial usage of CBRS PAL bands that cover 150MHz of spectrum ranging from 3.55 to 3.7 GHz [4].

To facilitate the implementation of the directives, Regulators have formulated leasing approaches, which define, for example, sets of rules, eligibility of lessors, lessees, pricing models, interference protection for lessors, and exclusion or protection zones such as airports, hospitals, and government premises [29]. In this context, the two most popular spectrum leasing approaches are the Universal Licensing System (ULS) [3] in the US and the evolved Licensed Shared Access (eLSA) in Europe [1]. The ULS and eLSA are both *based on a centralized database*, where lessees can query the available spectrum and choose a suitable lessor. To complete the lease, the parties then must agree on the terms and conditions of leasing and draft the corresponding contract. This process of negotiating contracts is conducted *manually* via meetings, emails, phone calls, and dialogues, which can easily take months [4]. The completed contract then needs to be signed by the contract participants, and the Regulator still has to approve it, which might, again, take weeks. Once approved, the Regulator updates the central database with the approved contract information in a non-real-time manner (e.g. the ULS spectrum database is updated once every 24 hours), further delaying the process. Finally, during the contract lifetime, the Regulator monitors and audits the contract information using third-party monitoring and auditing firms such as clearing houses.

Although it is possible to lease the spectrum using the ULS and eLSA, both suffer from a few key problems.

1) It is mostly manual process which means that it is slow, work-intensive, cumbersome, and unscalable to address the ever-increasing demand for spectrum usage for 5G, IoT, and Industry 4.0 use cases [1], particularly for short-term leases.
2) It is a centralized system, that requires trusting a third party to run and manage the service.
3) It does not support contract negotiation processes, which are typically performed outside the system manually via meetings, emails, phone calls, etc.
4) It does not provide an up-to-date listing of spectrum licenses, because the spectrum database is not updated in (near) real-time [3].
5) It does not provide anonymity to lessees' or lessors'. Anonymity is important since revealing either identity might negatively affect negotiations.
6) It does not protect lessees' operational privacy as any user in the system can trigger innocuous queries to the spectrum database and determine the types and locations of other lessees in a given region of interest, thus, compromising lessees' operational privacy [30].
7) Spectrum agreement auditing by the Regulator requires multiple intermediate parties, which is time-consuming and costly.

C. REQUIREMENTS FOR A STREAMLINED PRIVACY-PRESERVING SPECTRUM MARKETPLACE

From previous approaches and their shortcomings we can derive the key requirements for a streamlined spectrum leasing process. The requirements are divided into six categories, as follows.

1) Streamlining/Automation of the offer agreement process:
   a) The lessor must be able to create and update the leasing offers based on their needs/use-cases at any given time.
   b) The lessor and lessee must be able to negotiate the offer details (e.g., price) in near real-time.
2) User privacy:
   a) The lessor must be able to anonymously publish the offers, i.e., without revealing its real identity and only revealing information such as the frequency blocks in the offers.
   b) The lessor must be able to prove anonymously to the lessee that it is authorized to share its frequency block for the duration stated in the contract.
   c) The lessee must be able to prove to the lessor that it is authorized to choose an offer without disclosing its real identity.
3) Data confidentiality: Some information, such as price and the exact frequency band to be leased, must be shared only between leasing parties and eventually to the Regulator.
4) Secrecy until disclosure: Once the contract has been signed between the lessor and lessee, they must be able to prove their real identities to the Regulator.
5) Auditability without intermediaries: The Regulator must be able to audit offer and agreement information in near real-time without third-party intermediaries.
6) Trustworthy solution: Finally, the solution should not rely on a single entity for trust.

III. BACKGROUND

This section presents the main technologies used to build the automated spectrum leasing solution presented in this paper: decentralized identifiers, verifiable credentials, smart contracts, and Hyperledger Fabric.

A. DECENTRALIZED IDENTIFIERS (DIDS) AND VERIFIABLE CREDENTIALS (VCS)

Decentralized Identifiers (DIDs) are a privacy-preserving identifier technology that has recently received considerable attention. There are several different DID technologies in development [31], and although they started with different goals and solutions in mind, lately many of them have adopted the approach and format of the W3C DID specification [32] being developed by the Decentralized Identity Foundation [33], thus rendering them increasingly interoperable. A key aspect of DIDs is that they are designed not to be dependent on a central issuing party (identity provider) that creates and controls the identifier. Instead, DIDs are created

and managed by the identity owner (or a guardian on the owners behalf, if the owner does not have the capacity to manage their identifiers key themselves), an approach known as *Self-Sovereign Identity (SSI)* [34]. The DID specification [32] defines a DID as a unique string (prefixed by did and a string indicating a particular DID technology), often derived from the public key used with the identifier.

The DID is resolved to the associated DID Document, which may contain additional information about the holder of the DID, such as the associated public keys and, service entry points. The DID Document could be either self-contained as in the case of did:key [35] and did:self methods [36], or stored in a distributed ledger as in the cases of Sovrin [34] and Hyperledger Indy [37]. The General Data Protection Regulation (GDPR) in the EU and other similar legislation's have made storing personally identifiable information on a non-mutable platform such as a Distributed Ledger Technology (DLT)III-B problematic [38], so for this reason, Sovrin has already excluded individuals' DIDs from the ledger, and similar treatment may be applied to the DIDs of e.g., IoT devices if they reveal personal information about their owner or user.

In addition to identifiers, there is also a need for a mechanism to associate machine-verifiable properties with the identifier of an entity, for example, that the entity is a valid company registered in a certain country. Such an approach (analogous to traditional authorization certificates) in the language of the Decentralized Identity Foundation is known as a *Verifiable Credential (VC)* [39]. VCs are signed and structured digital documents created by a (trusted) issuer, and they can be used to provide proof of certain attributes of an entity, such as a person or a device [39]. They are designed to be to prove the contained attributes without revealing other sensitive personal information.

The DIDs and VCs are used together to enable secure and privacy-preserving identity management. DIDs can be used to identify the owner of a VC, whereas VCs can be used to prove specific attributes. For example, a university might issue a VC to a student stating that they have completed a particular degree program [39]. The student could then use their DID to prove that they are the owner of the VC, thus proving that they have completed the degree program, without revealing their name or other identifying information.

B. DISTRIBUTED LEDGER TECHNOLOGIES (DLT)

A Distributed Ledger Technology (DLT) is a peer-to-peer (P2P) network, where every peer has access to a shared state called a ledger [40]. The peers agree, following a consensus protocol, on how to update the ledger by inserting new records called transactions; however once inserted, the records cannot be modified or deleted, thus making the ledger *immutable*. Currently, the most common types of DLTs are the different blockchains.

Several different DLTs have emerged over the years, with each system having a unique set of strengths and limitations depending on its architectural characteristics and design choices. According to Wust et al. [40], DLTs can be classified based on who is allowed to read the data and who is allowed to write new data. In a *public* DLT, anyone can read the data, while in a *private* DLT only authorized parties have access to the data, so a public DLT provides transparency and non-repudiation of the data, while a private DLT provides privacy of the data. Similarly, in a *permissionless* DLT, anyone can write data (provided they meet the requirements of the consensus mechanism), while in a *permissioned* DLT only authorized parties can perform writes. Of the four possible types of DLTs, this classification, currently enables three types to be realized in practice. Bitcoin [41] and Ethereum [42] are examples of public and permissionless DLTs, whereas Hyperledger Fabric [7] is a private and permissioned DLT. Public and permissioned DLTs such as Hyperledger Indy [37] also exist, and they are well suited for cases where the DLT peers must be authorized and need to be identifiable, while maintaining the necessity for public verifiability.

Smart contracts [43] are another important feature provided by several DLTs: they are distributed applications executed on the ledger. Whenever an entity interacts with a smart contract, these operations are executed by all (full) nodes in the DLT network in a deterministic and reliable manner, one of which is selected to store the contract execution outcome (if any) in the ledger. Smart contracts can verify the DLT identities and digital signatures, perform general purpose computations, and invoke other smart contracts. The code of the smart contract is immutable and cannot be surreptitiously modified, not even by its owner. Moreover, because all transactions sent to a contract are recorded in the DLT, it is possible to obtain all the historical values of the contract. Smart contracts typically refer to code running on the Ethereum, but similar functionality is available in other DLTs. In particular, in the permissioned Hyperledger Fabric, such functionality is called chaincode, and simpler, more constrained scripts can also be run on Bitcoin.

Private data collection [44] is another feature provided by some DLT, such as Hyperledger. Typically, with a DLT, data are recorded on a shared ledger that is accessible to all participants. However, certain data within a transaction may be sensitive or confidential, making it undesirable to expose it to all participants in the DLT network. To address this concern, Fabric provides the concept of private data and offers features, known as private data collection to handle such scenarios. With this feature, an entity saves private data to its own private data collection and stores the hash of private data with a hash to the ledger. Now, the entity can selectively share private data with desired participants. These recipients can then generate the hash of the private data they receive and compare it to the hash stored on the public ledger. This verification process ensures that the shared data has not tampered with and can be trusted. This feature ensures the handling of private data in a private and secure manner.

IV. RELATED WORK

Researchers have proposed using blockchains and smart contracts to address issues with current spectrum leasing approaches, as discussed in Section II-B. Seppo provided high-level analysis on how blockchains and smart contracts can automate the process of spectrum access, where the blockchain could enable trust among participant and the smart contract could significantly reduce transaction costs by automating the complex business-to-business workflow of contracting and brokering [5]. However, this solution fails to meet the user privacy and negotiation requirements of the lessor and lessee (R1b, R2), as well as information sharing within the transacting parties (R3).

Rawat and Alshaikhi proposed a public blockchain (Bitcoin) solution that allows the mobile *virtual* network operator (lessee) to lease wireless network resources from the primary wireless resource-owner (lessor), based on service-level agreements (SLAs), without sharing their private information with anyone [45]. Here, every lease transaction is digitally signed by the leasing parties and recorded in the blockchain, which provides proof of wireless resources being leased by the lessee from the lessor, thereby preventing the lessor from over-committing its resources to the lessee. Additionally, the solution recommends the use of changeable public keys to maintain the privacy of leasing parties. However, the study did not elaborate on how leasing parties handle user privacy (R2). Moreover, if the SLAs terms were to change, updating the schema would require the complete redeployment of the solution, which is laborious and error-prone.

Pascale et al. proposed a permissioned blockchain (Hyperledger Fabric) based spectrum access system (SAS) that aims to address the trustworthiness of SAS administrators (Lessors) in the current SAS system [28]. The proposed solution allows various parties to participate in the spectrum access system and ensures that spectrum policies are strictly enforced without requiring the participants to trust individual SAS administrators, who are assumed to be trustworthy in the current SAS system but might not be trustworthy in reality. While this study utilizes similar technologies such as Hyperledger Fabric, to address the issue of trustworthiness, it does not address contract automation, negotiation, and privacy of the entities (R1, R2).

Liu et al. and Tu et al. focused on smart contracts to enable Small Cell-as-a-Service (SCaaS) [48] type services for individual users and retailers, which are provided for medium to large scale small-cell operators such as shopping malls, hospitals, etc [47]. The authors also leveraged smart contracts to demonstrate simple business agreements in the form of service level agreements between individual users, retailers, and mobile network operators, which simplified the process of agreements and reduced the cost of providing SCaaS. Similarly, Gorla et al. and Kim et al. proposed a blockchain-based spectrum platform, where lessors', lessees' and spectrum information are recorded on the blockchain and a smart contract is used for spectrum leasing. In these cases, lessees' privacy is an issue as lessees' transaction information recorded on the platform may contain lessee-specific information, which can be accessed by any node in the blockchain. Therefore, the privacy of the lessee cannot be guaranteed, and thus does not meet the anonymity requirements of the lessor and lessee (R2) as well as information sharing within the transacting parties (R3).

Liu et al. and Tu et al. recommend encrypting lessees' information prior to storing it to the blockchain. Although this approach safeguard lesses' privacy, it also involves additional steps to encrypt and decrypt information, which requires key management to achieve required confidentiality that is by no means a trivial task. Additionally, none of these studies tackled contract negotiation, data privacy, and regulatory compliance (R1b, R3, and R5).

The related work has focused on the concept of using blockchain and smart contracts for wireless resources, but they have a few shortcomings. First, there is concern regarding the privacy of both the lessor and lessee, as their personal information is recorded in the blockchain, which can be accessed by any user. Although some authors, such as Tu et al., suggest encrypting the data before storing it in the blockchain, this would require successful keys management. Second, most of the related work is based on a public blockchain, which means that the transaction information is public, and thus, available for any user. However, the spectrum market is highly regulated, and transaction information should only be available to authorized parties. Xiao et al. proposed using a permissioned ledger (Hyperledger Fabric), but they did not consider contract automation and actors' privacy concerns. Finally, none of the related works address contract negotiation between the lessor and lessee or the accessibility of transaction data, that is, that certain transaction data are private and should be only accessible to transacting members.

As none of the solutions successfully address all requirements, we describe an automated spectrum marketplace (ASM) that does so. For this, we use VCs and DIDs to ensure participants' privacy, smart contracts to automate contract agreement participants, and a permissioned ledger to enable trust and data privacy, as well as to simplify the auditability of spectrum transaction information by removing third-party clearing houses.

V. SOLUTION DESIGN

This section describes the architectural choices taken during the design of the automated spectrum marketplace (ASM) solution and answers the research question 1) *"How can the current mostly manual process of spectrum (sub)leasing process be automated to increase the efficiency, reduce the complexity, and time-consumption?"*. The focus has been on meeting the identified requirements while providing privacy, efficiency, and streamlining the spectrum leasing process. This section highlights the key steps of the spectrum leasing process; however in reality the process can be more complex and may require additional interaction between participants.

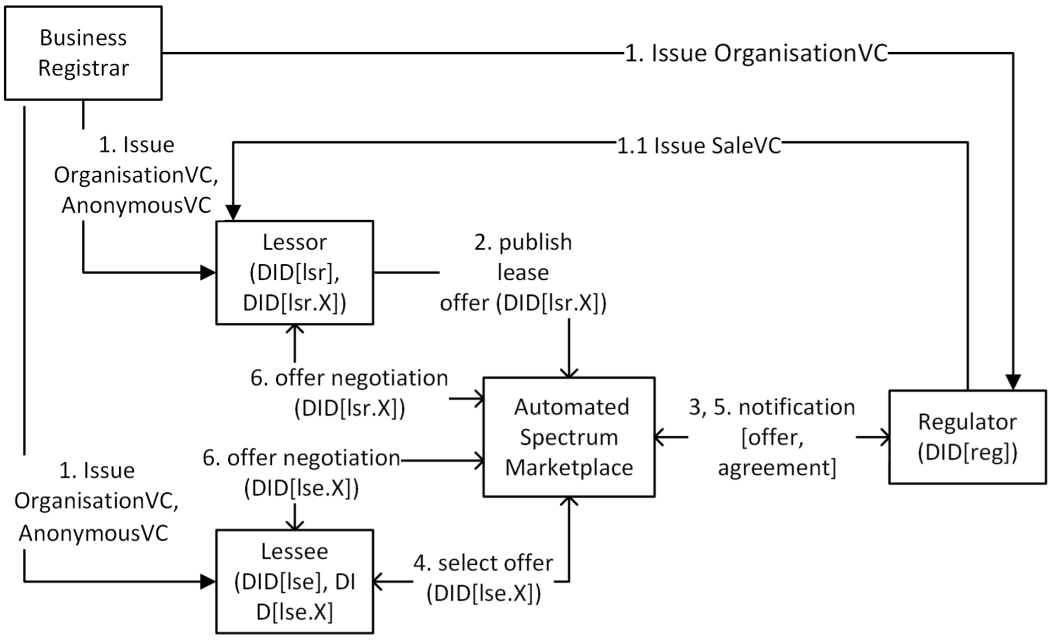

FIGURE 2. Actors interaction with ASM.

Support for additional interactions can be added to the solution when necessary.

A key feature of the design is a decentralized marketplace that automates the spectrum leasing process while ensuring user privacy, data confidentiality, and auditability. A permissioned ledger is used to store transaction information and to ensure transparency and trust. However, certain sensitive data such as price and exact trading frequency, are kept secret using private data collection. To ensure user privacy in the marketplace, the design utilizes ephemeral DIDs as anonymous identifiers and then VCs for authorization.

Figure 2 illustrates how different actors interact with ASM. It is operated by a consortium, mainly consisting of spectrum regulators and lessors (operators), optionally lessees (enterprises). Due to the highly regulated nature of the spectrum market, the spectrum regulator acts as the coordinator responsible for ensuring that the participants follow the regulations and for handling potential disputes. The consortium sets the rules of membership, eligibility for spectrum leasing, and data access both for the actors and the public. An example policy is such that anyone can access/read the spectrum leasing offers, but to participate in the spectrum leasing process, the entity must be a registered company and its identity reviewed by the spectrum regulator.

A Business Registrar (government authority that maintains business/company registration) issues VCs to all participants (step 1 - Figure 2). With these VCs, the lessor and lessee prove to the regulator that they are registered organizations, and the regulator then grants them access to the marketplace. Similarly, the regulator issues a VC with spectrum information to the lessor (step 1.1 - Figure 2). This VC indicates the lessor's spectrum license ownership. The lessor can now create new leasing offers and anonymously publish them to the marketplace using anonymous VCs (step 2 - Figure 2), which rely on ephemeral decentralized identifiers. The lessee can then select suitable offers to bid on (step 4 - Figure 2). Both participants negotiate on the price (step 6 - Figure 2), and once agreed, the offer agreement is recorded on the ledger by the marketplace, which can be audited by the regulator in near real-time. The marketplace notifies the regulator of the published offers and agreements (step 3,5 - Figure 2).

To satisfy R1 (**automation of the offer agreement process**), the design uses smart contracts. These smart contracts allow lessors to create new offers at any given time, either by defining new attributes to cater to specific use cases, or by using pre-existing leasing offer templates provided by other lessors. Regardless of whether the offers are newly created or based on existing ones, certain fields in the offer are mandatory, such as status and validity period. The status field indicates the state of the offer/contract. When an offer is newly submitted, its status is set as *pending*. If the marketplace or authority approves the offer, status transitions to *valid*. Once the offer is concluded, the status becomes *active*. Finally, when the contract period expires, the status transitions to *concluded*. However, if the offer is not agreed upon by either party, the status is marked as *cancelled*. This approach offers flexibility to lessors e.g., on the number of parameters included in the contracts, allowing them to create new kinds of offers without requiring any changes to the solution, thereby catering to future use cases.

Furthermore, the offer includes built-in negotiation logic through smart contracts, enabling the lessor and lessee to engage in near real-time negotiations and eventually reach an agreement. The details of the contract agreement is recorded on the permissioned ledger that is distributed across the consortium nodes. This decentralized approach transfers the

required trust from individual entities to the consortium, enhancing the overall trustworthiness of the solution, thus addressing the requirement **trustworthy solution** (R6). In addition, the ledger provides immutable proof of offer transactions, which can be directly audited by the regulator in near real-time without requiring third-party intermediaries, thereby addressing the requirement of **auditability without intermediaries** (R5).

To support the logic of offer creation and negotiation, the design utilizes two smart contracts (chaincodes): the *lease-query* and the *lease*. The *lease-query* operations enable read, while *lease* operations enable write to the ledger and private data collections.

The lease offer is executed on the permissioned ledger that stores all the information including business sensitive data i.e., price and exact frequency to the ledger, as accessible by all the participants. However, this does not suffice to satisfy the requirements of **data confidentiality** (R4). One solution is to use a separate database to store and share secret information only with transacting participants. However, this requires integration with the distributed marketplace. Additionally, the proof of the secret information should be stored in ledger to ensure data integrity. Permission ledgers, such as Hyperledger Fabric [44], already support private data collection by design, which makes it suitable for addressing this requirement.

Another shortcoming of the permissioned ledger is that it compromises the **users' privacy** requirements (R2), for same reasons mentioned earlier. All user information is stored in the permissioned ledger and remained accessible to all participants. One approach to address user privacy is to encrypt user information before adding it to the ledger, as suggested by Liu et al. and Tu et al. However, as mentioned in Section Related Work IV, this adds overhead, and disclosing limited information to the participants would require additional implementation logic. Alternatively, supporting users' privacy by providing limited resource access to third-party protocols such as OAuth2 [53] could be considered. However, OAuth2 is better suited for centralized solution and fall short of meeting the trustworthiness requirements of a distributed marketplace. Thus, the ASM design uses decentralized identifiers (DIDs), to address privacy requirements.

To ensure participants anonymity during the agreement process, the design incorporates the use of short-lived ephemeral DIDs. These DIDs are generated by the lessors (DID[lsr.X]) and lessees (DID[lse.X]) respectively, and are *unique for each transaction*. These DIDs are called Anonymous Identifiers and are listed in Table 1. In addition, to allow participants to disclose their official identities to each other after agreement has been reached, the design uses Public Identifiers. These identifiers are long-term DIDs generated by the regulator (DID[reg]), lessors (DID[lsr]), and lessees (DID[lse]) respectively. Table 1 summarises the actors' DIDs.

The DIDs themselves are not sufficient to prove that the participants are authorized to participate in the marketplace. For this purpose, the design uses verifiable credentials (VCs). Table 2 summarizes the different types of VCs used in the ASM. To meet this requirement, **authorized to share the spectrum anonymously** (R2b), the lessor uses SaleVC, which is issued to them by the regulator. The SaleVC contains an anonymous identifier and spectrum metadata determined by the lessor and includes fields such as frequency range, coverage area of the spectrum, and lease duration. Similarly, to meet the requirement **authorized to choose an offer anonymously** (R2c), the lessee uses the LesseeVC, which is issued to them by the Business Registrar. It contains lessees' anonymous identifiers and their roles. With LesseeVC, a lessee can anonymously select an offer from the marketplace and prove that it is authorized to participate in the marketplace.

Further, to meet the requirement of **secrecy until disclosure** (R4), the Business Registrar issues OrganizationVCs to all actors (lessor, lessee, and regulator). These VCs contain the public identifiers (long-term public DIDs) of the respective actors, their names, business ID, and roles, which indicate whether the VC holder is a regulator, or leases the spectrum. The actors can use these VCs to prove their official public identities to other participants and that they are legitimate organizations authorized by the Registrar to participate in the marketplace.

Finally, the lessee proves their spectrum lease to the regulator with a LeaseVC issued by the Lessor after the negotiations have concluded. The LeaseVC includes the anonymous or public identifier of the lessor and lessee (if they agree to reveal their identity) and spectrum metadata.

A. AUTOMATED SPECTRUM MARKETPLACE OPERATIONS

The steps illustrated in Figure 3 describe the detailed operations of the Automated Spectrum Marketplace (ASM), hereafter also marketplace. To simplify the process, certain assumptions were made. First, each actor involved has been issued OrganizationVCs by the Business Registrar, along with AnonymousVCs to the lessor and lessee. Second, the regulator issued a SaleVC to the lessor. Third, the solution ensures that participants receive notifications regarding relevant events such as a new offer published. Finally, consortium members do not misbehave and can be trusted.

1) The lessor creates an offer, that can be updated at any time until the Step 7.
2) The lessor anonymously submits the offer on the marketplace.
3) The marketplace automatically reviews the lease terms of the offer.
4) If the offer violates the lease terms, the marketplace notifies the regulator for further review.
5) The Regulator checks the VC, if the VC verification fails, or the offer does not meet the regulatory requirements, then the regulator sends a deny publish message to the marketplace.
6) The marketplace forwards deny publish message to the respective lessor.

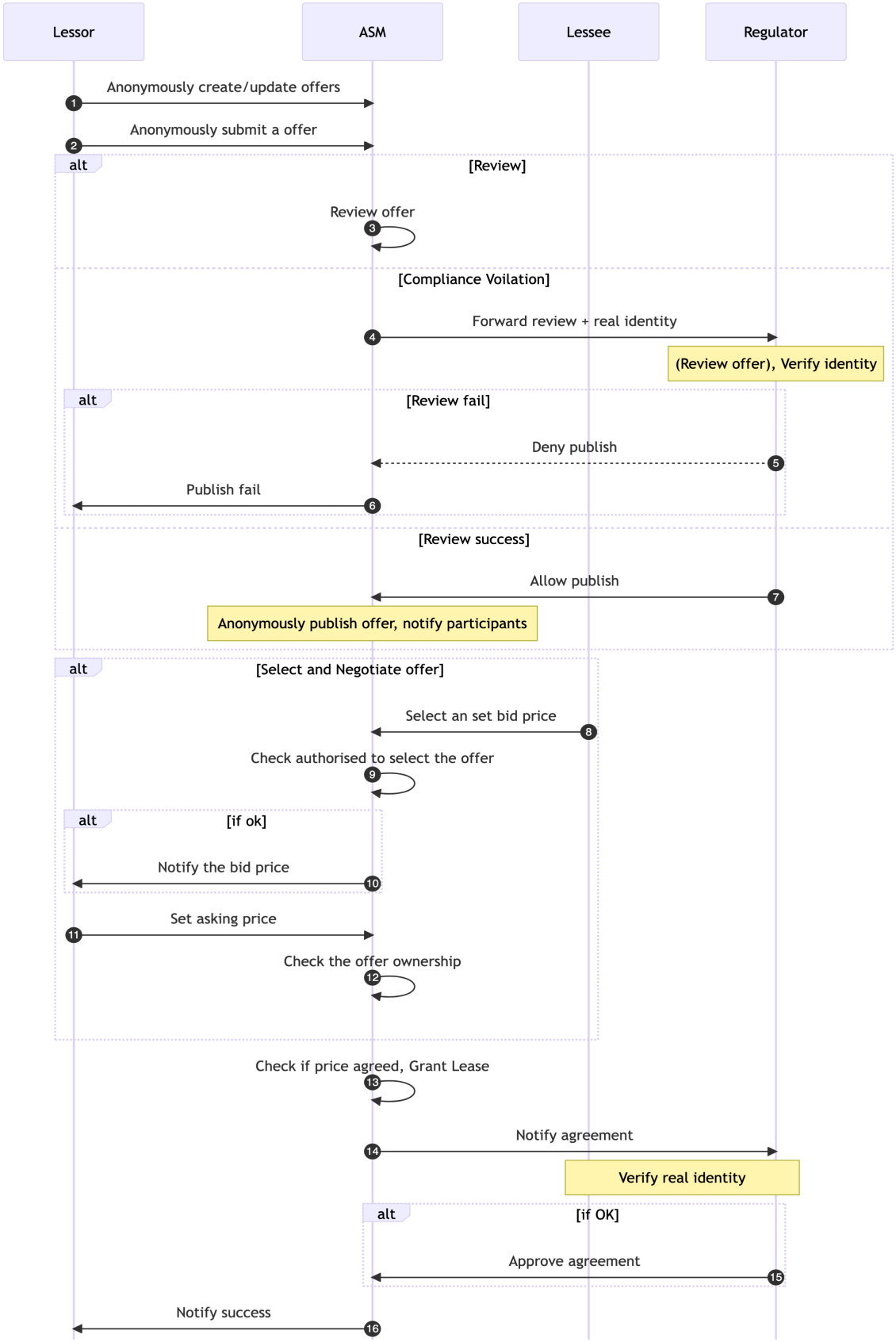

FIGURE 3. Details of the spectrum leasing process between the lessors, lessees, regulator, and the marketplace.

TABLE 1. DIDs for the actors.

Identifiers	DIDs	Actors	Purpose
Public Identifiers	DID[reg], DID[lsr], DID[lse]	Regulator, Lessor, Lessee	Provide Official identifiers for all the actors
Anonymous Identifiers	DID[lsr.X], DID[lse.X]	Lessor, Lessee	Provide anonymous identifiers for the Lessee and Lessor

TABLE 2. Summary of verifiable credentials.

Credentials	Purpose	Issuer	Holder	Verifier	Anonymous	Identifiers	Content
OrganizationVC	Identify all actors	Business Registrar	Regulator, Lessor, Lessee	Anyone	No	Public Identifiers	Name, Business ID, role (lessor, lessee, regulator)
LesseeVC	Anonymously prove that the lessee is authorized to participate in the marketplace	Business Registrar	Lessee	Lessor	Yes	Anonymous Identifiers	role (lessee)
SaleVC	Anonymously prove that the lessor has the spectrum	Regulator	Lessor	Lessee	Yes	Anonymous Identifiers	spectrum metadata*
LeaseVC	(Anonymously) Prove that the Lessee has leased the spectrum	Lessor	Lessee	Anyone	Optional	Anonymous Identifiers	spectrum metadata*

7) Once the verification and review of the offer data are successful, the regulator sends an OK message to the marketplace. Subsequently, the marketplace publishes the offer and shares it with each member.
8) The lessee checks or queries the marketplace regarding list of offers. From this list, it can select the most relevant offer and set a bid price using LesseeVC.
9) The marketplace checks whether the lessee (presenting LesseeVC) is authorized to select an offer.
10) If the lessee's anonymous verification is ok, the marketplace notifies the lessor of a new bid.
11) The lessor sets an asking price and sends anonymous proof of offer ownership to the marketplace (SaleVC).
12) The marketplace validates the lessor's anonymous proof.
13) The marketplace validates whether the transacting members agree on the same price or not. If there is an agreement on the price, the marketplace grants the lease.
14) The marketplace notifies the regulator of this agreement. The regulator verifies the lessee's real identity.
15) The regulator sends its approval of the agreement to the marketplace.
16) The marketplace then sends a success message to the lessor. Finally, the lessee can lease the spectrum for a specific period.

VI. IMPLEMENTATION

This section presents the prototype of the ASM design. The prototype was implemented using two main components: Hyperledger Fabric [7] as the permissioned blockchain (because it supports private data collection [44]), and self-sovereign identities (SSI) based on the Horizon 2020 IoT-NGIN framework [8]. Although Fabric and IoT-NGIN were chosen for ASM implementation, the ASM design is not tied to these systems, as it can be developed and deployed using any other blockchain and self-sovereign identity solution that supports the necessary functionalities. Section VI-A describes the details of the smart contracts (chaincode) implementation, while Section VI-B describes how the identity operations have been implemented with the IoT-NGIN. The prototype was tested to demonstrate the feasibility and effectiveness of the ASM system.

For simplicity, this implementation uses a Fabric setup provided by hyperledger fabric-sample repository [54] for deploying the Fabric network. In this setup, there is a Lessor, Lessee, and Regulator. However, for real-world deployments, it is recommended to have multiple Regulators, Lessors, and Lessees to meet trust requirements. In the future, the ASM system can be further developed and tested in larger deployment scenarios to evaluate its scalability and resiliency. Additional features, such as notifications system and dedicated portals for users, can be implemented to enhance the usability and convenience of the system.

A. CHAINCODE

The chaincode was implemented using Go programming language [55] version 1.19. The chaincode was deployed to the Fabric network [7] version 2.4, which executes the chaincodes and stores the transactions in CouchDB [56] version 3.1.

To optimize the system, the offer operations were divided across two chaincodes: *lease-query* and *lease*. This division helped organize the functions and reduce the number of chaincode installations and approvals. Further, CouchDB was used for customized queries and Private Data Collection was used to store the private data of the participants, thus reducing the administrative overhead of setting the state database and collections.

The *lease* contract includes an offer schema as shown in Listing 1. The ID is the unique key, with which the offer can be queried from the ledger. The name field contains human-readable offer names. The lessor-id and lessee-id

fields contain anonymous DIDs of the respective participants (the lessee-id value is only inserted or updated after the offer has been agreed upon). The status field indicates the current status of a lease agreement. The duration field specifies the offer's validity period. The spectrum-metadata field includes metadata related to the spectrum, including the frequency range, channel block, and location. The frequency range is the range of frequencies, that the lessors' have purchased from a Regulator (this field must match the frequency range in SaleVC). The channel block is a block of spectrum available for leasing. In this case, a fixed C block is used for simplicity. The location is the coverage area of the spectrum frequency.

In addition to the offer schema, the *lease* contract includes fields for price and frequency. The price field is used by the lessor to set their asking price and by the lessee to set their bidding price. Both parties store their price values in their respective private data collections. The lessor uses the frequency field to specify the exact frequency that they want to lease. This value was stored in the lessor's private data collection.

```
{
  "ID": <a unique identifier>,
  "name": <name of the offer>,
  "lessor_id": <anonymous did of the lessor>,
  "lessee_id": <anonymous did of the lessee>,
  "status": <current offer status>,
  "publish": <allow or deny>,
  "spectrum-metadata": { // aka spectrum metadata
    "frequency_range": <range owned by the lessor>,
    "channel_block": <spectrum block>,
    "location": <geocode of spectrum coverage area>
  },
  "duration": <agreement validity period>
}
```

Listing 1: Fields of the offer schema.

The fields of Offer Schema 1 were simplified for the prototype. In a real-world implementation, the lessor might have offers that have fewer or more fields than those presented here, depending on their lease offer needs, based on which they would be able to create new schemas to suit their own use cases.

The offer schema is called by various functions in several steps of the agreement process, as described below:

SaveOffer - This function implements the first step of ASM operations, as described in Step 1. It allows the lessor to create a new lease offer at any given time, using the input parameters and saving it to the ledger. It also saves the offer's exact frequency in the lessor's private data collection.

UpdateOffer - This function allows the lessor to make changes to an existing offer and update various parameters of the offer, such as the frequency to be leased and, the lease duration. As with *SaveOffer*, the values cannot be changed once the offer has been reviewed and approved by the regulatory.

ReviewOffer - It implements review steps 3 - 7. To simplify the implementation, the automated review by the marketplace in Step 3 is ignored in favor of regulator review. The execution of this function is limited to the Regulator, who reviews the offer and decides whether to permit or reject it, taking into account the leasing regulation, such as determining if the spectrum band is not allowed to be shared by the lessor. This function allows the Regulator to update the value of the *publish* parameter in the offer schema to reflect their decision, which allows the lessor to advertise their offer to the marketplace. It is important to note that the Regulator's decision is final, and the offer cannot be modified once it has been reviewed and approved.

LessorAskingPrice - This function implements step 11 of the ASM operations, which enables the lessor to set the asking price for their offer. The specified price is saved to the lessor's private data collection, while the salted hash of the price is stored in the ledger. The lessee requests information about the asking price from the lessor, and once the lessee receives the asking price with the corresponding salt from the lessor, the lessee can compare it with the salted hash on the ledger to validate the asking price.

LesseeBidPrice - It implements Step 8 of the ASM operations. It has similar functionality as the *LessorAskingPrice*, with the difference that the lessee can set the bid price that is saved to their private data collection.

AgreeToLease - This function implements 13, which is executed each time a lease or bid price is set. This function verifies whether the lessor and lessee have agreed on the same price, by comparing the hash of asking and bid prices of the lessor and lessee from the ledger. If the comparing hash value matches, it means that both parties have come to an agreement otherwise, either party can propose a new price until agreement, and the next step can be followed.

GrantLease - Implements the latter part of Step 13 that ensures that the lessor has been granted access to the spectrum frequency for the agreed duration. It re-evaluates that the transacting parties have agreed on the same price, and if yes, the function adds the lessee-id, and updates the agreement status to *active*. Finally, the price agreements are deleted from both the participant's private data store and a receipt is created in each private data store.

Overall, the chaincode implementation in the ASM prototype facilitates the automation of leasing contract agreements between participants, ensuring transaction auditability in near real-time and without a third-party. It also enables secure storage and sharing of sensitive information related to the spectrum leasing process.

B. SELF-SOVEREIGN IDENTITIES

Identities were implemented using Python 3.8. They leverage two key functionalities offered by the IoT-NGIN library: the generation and display of decentralized identifiers (DIDs) and the creation and verification of verifiable credentials (VCs).

There are four types of VCs: OrganizationVC, LesseeVC, SaleVC, and LeaseVC. Listing 2 shows the schema used to create the Organization and Lessee VC. The schema has two main fields: *issuer* and *credentialSubject*. Note that other compulsory fields of credentials such as issuance and

expiration dates are omitted for simplicity. The *issuer* field has the Business Registrar's DID, which denotes that the credentials have been issued by the Business Registrar. The *credentialSubject* has the following sub-fields: *id*, *role*, and *businessId*. The *id* is the long-lived DID of the target subject, subject's role is, in this case regulator/lessor/lessee, and the businessId is the official business ID of the subject. Both the LesseeVC and OrganizationVC have the identical fields, differing only in the lifespan of the credentialSubject id (short-lived for LesseeVC and long-lived for OrganizationVC). In the case of AnonymousVC, businessId is omitted to preserve the subject's anonymity.

```
{
"issuer": <Business Registrar's DID>,
"credentialSubject": {
  "id": <DID of the subject, either short or long-lived>,
  "role": <regulator/lessor/lessee>,
  "businessId": <business ID of the entity, omitted for
  ↪  anonymous credentials>
}
}
```

Listing 2: Organization/Anonymous VC schema.

Listing 3 shows the schema used to create SaleVC and LeaseVC. The schema has two main fields: *issuer* and *credentialSubject*. The *issuer* field denotes who has issued the VC, while the *credentialSubject* field denotes to whom the VC has been issued. In the case of LeaseVC, the *issuer* is the Regulator DID, while with LeaseVC, the issuer is a short-term DID of the Lessor. Similarly, with SaleVC, the *credentialSubject.id* is the short-term DID of the Lessor. Additionally, the *credentialSubject* has *spectrumMetadata*, that contains spectrum specific values and has sub-fields: *frequencyRange*, *channelBlock*, and *location*. The *frequencyRange* field specifies the frequency range owned by the Lessor, *channelBlock* specifies the spectrum block available for sale/lease, and *location* specifies the coverage area of the spectrum frequency.

VII. RESULTS

This section presents the results of the experiments performed on the ASM prototype. The key metric is the latency of the end-to-end offer agreement between the lessor, lessee, and regulator.

The offer transaction history used by the regulator for auditing transactions was also considered in the measurement. Section VII-B describes the measurement results of the latency and evaluates whether the ASM improves the performance compared with the current leasing solutions.

The throughput of the offer agreement was not used in the metrics. The reason for this is that the volume of transactions remains limited because of the limited number of frequency bands being traded. Similarly, the latency and throughput of identity creation were not measured because user identities are typically created once. Thus, the frequency of identity and verifiable credential transactions is not very interesting, because there are only a few identity transactions.

```
{
"issuer": <Regulator's DID for sale credential, short-term DID
↪  of the lessor for lease credential>,
"credentialSubject": {
  "id": <short-term DID of the lessor for sale credential,
  ↪  short-term DID of the lessee for lease credential>,
  "spectrumMetadata": {
    "frequencyRange": <frequency range>,
    "channelBlock": <channel block>,
    "location": <location>
  }
}
}
```

Listing 3: SaleVC/LeaseVC schema.

A. EXPERIMENT SETUP

The experiments were performed on a virtual machine with Kernel version 5.4.0, Ubuntu 20.04 Operating System, an Inter(r) Xenon CPU@2.70 GHz, 4GB of RAM, 4 virtual CPUs, and 60GB of disk.

TABLE 3. Latency of offer agreements.

TPS	Max Latency(s)	Min Latency(s)	Avg. Latency(s)
50	43.89	26.74	32.20

For the measurement, a fabric network was created using Hyperledger Fabric v2.4 [7] and, CouchDB v3.1 [56] as the state database was deployed in Docker containers v20.10 [57], and Golang v1.19 [55] was used for chaincodes. For network deployment, test-network scripts from the Fabric sample [54] were customized and additional bash scripts were added to automate deployment of three organizations representing a lessor, lessee, and regulator, respectively, where each peer had their respective instance of CouchDB to store the transaction states and Private Data Collection to store private data. Furthermore, the setup allowed addition or removal of an organization to the existing network without halting the network, which ensured that there was no downtime for offer agreement.

B. LATENCY OF OFFER AGREEMENT

Although a dedicated bench-marking tool such as Hyperledger Caliper [58] exists, it was not considered for latency measurement because of the simple test cases and low volume (50) of transactions per second. Instead, a bash script is utilized to carry out several lease executions. The script invokes chaincode functions, involving 11 write and 17 read operations per transaction.

With measurement, only the technical part of the offer agreement was considered, meaning the total time taken to complete the offer agreement operations, which includes the, time to create an offer, approve the offer, publish the offer, set the offer asking and bidding price, negotiate on the offer price, and finally agree on the offer as well as query to obtain offer transaction history. Here, the time needed for non-technical part such as agreement review by the regulator and price negotiation between the lessor and lessee are then added on top of the technical time as it is upto the regulator, lessor, and lessee on how promptly/slowly they agree on the steps.

Table 3 presents the latency for transaction rate. The results demonstrate that the ASM can handle very a high demand of spectrum leasing and yet maintain an efficient performance. The outcomes indicate that ASM successfully executed 50 transactions per second, and the maximum latency for the offer agreement was below one minute. This is significantly faster than the currently used solutions, which can take days or weeks to complete similar transactions. Adding the times for non-technical aspects means that many agreements would not be reached within one minute, but even that time is possible if the lessor and lessee quickly agree on the price and the regulator's review process is automated, as discussed earlier.

VIII. ANALYSIS

This section details how the ASM design satisfies all of the requirements listed in Section II-B. It also answers the research question *How can the privacy of participants be protected during the spectrum leasing process?* (first four paragraphs) and *How can the auditability of spectrum leasing transactions be ensured in a transparent and verifiable manner?*

A. AUTOMATION OF THE OFFER AGREEMENT PROCESS (R1)

The ASM design effectively addresses the need for automation by leveraging chaincode to streamline the manual offer agreement process of offer creation/update and, negotiation, which otherwise can take weeks and, months to complete. Instead, the process can be completed within minutes. With ASM, lessors can use the chaincode to create their own customized contract templates with parameters specific to their needs/use-cases, thereby addressing requirement 1a. While new templates are not created frequently, there may be instances where updates to existing templates are required because of changes in contract requirements or to support new updated contractual terms. With the current spectrum leasing practices II-B, the process requires manual interactions with other parties. However, the ASM system is designed to support dynamic schemas and parameters using chaincode, which enables lessors to create or update offers in near real time. In addition, the chaincode includes functions to support near real-time negotiation between the lessor and lessee, thereby addressing requirement 1b. Both parties can set their desired prices privately and negotiate until they agree on a mutually satisfactory price. Once an agreement is reached, the chaincode can facilitate the creation of a legally binding agreement by recording the digital signatures of both parties in the blockchain. This streamlined approach simplifies the process, enhancing efficiency, automation, and convenience for all participants involved, ultimately achieving the leasing agreements within minutes at a low cost compared to current solutions II-B, which take multiple weeks or even months and often cost tens of thousands of dollars if not more. While some of the ASM operations, such as reviewing the offer by the regulator and the offer negotiation, are left to participants' discretion, the ASM significantly reduces the time and cost of offer agreement, making the cost of spectrum leasing drastically cheaper than traditional solutions.

B. USERS PRIVACY (R2)

To protect users' privacy, the ASM design enables participants to use anonymous decentralized identifiers (DIDs). Anonymous decentralized identifiers ensure that the participant's identity is not revealed unless they choose to reveal it themselves, and using multiple anonymous addresses helps maintain a high-level of user privacy. DIDs in conjunction with VCs enable participants to advertise information about their offers to other participants, but this information is vague enough and only includes details that are necessary for the lessee to make an informed decision. To prove that the information is owned by the lessor, the lessor uses anonymous verifiable credentials linked to the lessors' anonymous decentralized identifier issued by the regulator. The anonymous VC does not disclose the real identity of the lessor and only shares relevant information, which ensures the lessor privacy, thereby addressing requirement 2a. Further, to prove that the lessor is authorized to lease the frequency block for the leasing period stated in the offer to the lessee, the lessor utilizes its anonymous VC. This addresses the requirement 2b. Similarly, with the lessee's anonymous verifiable credentials, which ensures that the lessee is authorized to choose and offer without disclosing its real identity, thereby addressing the requirement 2c.

Additionally, to meet the **Secrecy until disclosure** (R4), the design uses anonymous DIDs and public DIDs. The anonymous DIDs are unique to each transaction, which eliminates the possibility of identity correlation while providing proof of identity. Furthermode, the digital signature enables the verification of the identity associated with the identifier, ensuring the authenticity and integrity of the transaction data. Furthermore, once an offer is accepted, participants reveal their real identifiers to each other via public DIDs that provide their real identity. Thus, this approach enables participants to maintain their secrecy until the point of disclosure thereby addressing requirement 4.

C. DATA CONFIDENTIALITY (R3)

To ensure that sensitive data are shared only among authorized participants involved in the offer negotiation process, the design uses the private data collection feature of Fabric. This feature stores the participants data off-chain, while maintaining a reference on the blockchain. This mechanism enhances data confidentiality by limiting access to only those with necessary permissions. It provides a secure way to share sensitive information, reducing the risk of unauthorized access and data leakage, and preserves privacy. By separating private data from the publicly visible data within blockchain consortia, participants can maintain control over their confidential information while still benefiting from the transparency and immutability of the blockchain.

Thus, this ensure the data confidentiality thereby addressing requirement 3.

D. AUDITABILITY WITHOUT INTERMEDIARIES (R5)

To achieve auditability without the involvement of intermediaries, the design incorporates Hyperledger Fabric [7]. The Fabric securely stores the digital signatures of all parties and creates an immutable record of the agreement on the ledger. This ledger not only stores digital signatures but also includes additional metadata such as lease duration and, frequency range. The regulator can access this ledger in near real-time, enabling them to audit the transactions effortlessly without the need for third-party intervention. This approach effectively addresses requirement 5, ensuring transparency and accountability throughout the leasing process.

E. TRUSTWORTHY SOLUTION (R6)

To ensure that the solution is trustworthy, the design uses the blockchain technology. By leveraging blockchain, data are distributed across multiple nodes and recorded on an immutable ledger. This decentralized approach ensures that the solution is not centralized and elimintes the need to trust a single entity. By implementing this methodology, the solution becomes more trustworthy and effectively addresses requirement 6.

F. THROUGHPUT

The prototype results demonstrate that the ASM is capable of handling a very high demand for spectrum leasing. To put this in perspective, currently in the US, spectrum leasing is widely practised compared to other countries. There are seven priority access licenses granted by regulatory authorities for each of the 3233 counties for a total of 22,631 licenses nationwide [59]. Assuming even worse case that each county has five lessees (i.e., small-medium businesses) per license, and each lease only lasts one hour (highly unlikely, as leasing time would be easily hours like for sports events), the total transactions nationwide per hour in this scenario would be 22631*5=113,151. Comparing this to ASM's transaction rate, which is 50 transactions per second, that is 3600*50=180,000 per hour, the ASM is easily able to handle this highly unlikely volume of leasing transactions with the prototype implementation.

G. SECURITY AND TRUST

The design presents certain security considerations. In contrast to centralized system, where malicious code could more easily hide within the system due to the consortium inaccessibility, the ASM's design utilizes smart contracts visible to the entire consortium. This transparency allows any partner to potentially to check contract changes, with the regulator systematically reviewing all new contracts. Consequently, the ASM significantly reduces the risk of malicious code presence. Furthermore, the regulator functions as a gatekeeper, granting access only to the authorized participants to prevent fraud and unauthorized actions, thereby enhancing the trust and system security. Furthermore, while the ASM design enables participants anonymity towards each other, the regulator can pierce the anonymity when necessary and is therefore able to detect misuse. Additionally, the immutable ledger provides accountability, thus incentivising proper behaviour amongst the actors and allowing swift correction of improper behaviour.

IX. DISCUSSION

The Automated Spectrum Management(ASM) system addresses the key problems of current spectrum leasing solutions by automating much of the end-to-end leasing process while providing better user privacy. Other solutions discussed in Section IV use blockchain technology such as Ethereum for spectrum access, but they do not address the end-to-end automation and privacy issues addressed by the ASM. Moreover, some solutions suggest encrypting user data before inserting them into the blockchain to ensure privacy. However, this requires the successful management of all associated encryption keys to achieve the required confidentiality [51], [52]. In contrast, the ASM uses anonymous decentralized identifiers (DIDs) and verifiable credentials (VCs) to ensure user anonymity, and the Hyperledger Fabric platform's private data collection to store users' private data. The ASM's smart contract also automates much of the offer agreement process, and the permissioned ledger used in the system enables easier regulation. Additionally, the ASM system has a feasible deployment process, because it does not require a complex setup, and network nodes can be easily added or removed without disrupting the network.

By widely adopting the ASM system, the two significant issues concerning short-term spectrum leasing for small and medium-sized companies discussed in Section II-A can be effectively addressed. First, the challenge of efficiently acquiring access to licensed spectrum can be mitigated, as the ASM enables businesses to easily lease spectrum for short-term within minutes. This streamlined process would likely make more spectrum available for leasing, opening up new business opportunities for lessors and lessees. Second, the privacy of users involved in spectrum leasing can be ensured as the ASM system provides business with the assurance of privacy and data confidentiality, safeguarding their valuable trade secrets. Furthermore, the ASM system is flexible and can support the creation of completely new contracts for new use cases with minimal or almost no changes, thereby making it well-suited for future use cases, allowing it to adapt to changing needs and demand for spectrum leasing.

X. CONCLUSION

Using the current spectrum leasing solutions for the short-term, leasing is unviable because of manual, complicated, and time-consuming contract agreements, negotiations, and costly auditing processes, along with the participants' privacy concerns.

The Automated Spectrum Marketplace (ASM) solution, described in this paper, makes prompt short-term spectrum leasing viable, by leveraging dynamic contract creation, negotiations, and auditing through DLT, and addressing the privacy of the user via decentralized identifiers (DIDs), verifiable credentials (VCs), and private data collections. The prototype implementation of the ARM solution demonstrates a significant improvement in the speed of the spectrum agreement process ensuring the participants privacy, compared to current solutions.

This domain holds promising opportunities for future research. Further research is required to refine and implement the solution within a mobile network and test its capabilities. For instance, to predict spectrum demand and optimize spectrum allocation for more effective spectrum leasing, technologies such as ML/AI can be incorporated. In addition, a more comprehensive evaluation of the proposed automated solution in terms of scalability, security, and real-world adaptability can be undertaken. Moreover, to achieve widespread impact requires collaborative standardization efforts among operators, industry stakeholders, and regulators to ensure the interoperability of the solution. Further real-world deployment and validation of the solution within testbeds and pilot projects will provide valuable insights and pave the way for its practical adoption.

ACKNOWLEDGMENT

The authors would like to thank Prof. Valtteri Niemi and Finn Pedersen for their valuable comments.

REFERENCES

[1] *Reconfigurable Radio Systems (RRS); Evolved Licensed Shared Access (ELSA); Part 2: System Architecture and High-Level Procedures*, document TS 103 652-2-V1.1.1, ETSI, Jan. 2020.

[2] M. D. Markus, S. Srikathyayani, and B. Biljana, "Spectrum sharing technology | LSA and SAS white paper," Intel, Santa Clara, CA, USA, White Paper, Oct. 2015. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/spectrum-sharing-lsa-sas-paper.pdf

[3] *Universal Licensing System*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.fcc.gov/wireless/universal-licensing-system

[4] "Promoting efficient use of spectrum through elimination of barriers to the development of secondary markets," Federal Commun. Commission, Washington, DC, USA, Tech. Rep. 04, Sep. 2004. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-04-167A1.pdf

[5] Y. Seppo, "Analysis of blockchain use cases in the citizens broadband radio service spectrum sharing concept," in *Proc. Int. Conf. Cogn. Radio Oriented Wireless Netw.*, Jan. 2018, pp. 128–139, doi: 10.1007/978-3-319-76207-4_11.

[6] "Decision of the president's chamber of the bundesnetzagentur," Bundesnetzagentur, Germany, Tech. Rep., 2018. [Online]. Available: https://www.bundesnetzagentur.de/SharedDocs/Downloads/EN/Areas/Telecommunications/Companies/TelecomRegulation/FrequencyManagement/ElectronicCommunicationsServices/FrequencyAward2018/20181214_Decision_III_IV.pdf?__blob=publicationFile&v=2

[7] *Hyperledger Fabric*. Accessed: May 9, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/blockchain.html

[8] *H2020 IoT-NGIN/Enhancing IoT Cybersecurity and Data Privacy/Privacy Preserving Self-Sovereign Identities*. Accessed: May 9, 2023. [Online]. Available: https://gitlab.com/h2020-iot-ngin/enhancing_iot_cybersecurity_and_data_privacy/privacy-preserving-self-sovereign-identities

[9] "Spectrum leasing in the 5G era," GSMA, Tech. Rep., Jan. 2022. [Online]. Available: https://www.gsma.com/spectrum/wp-content/uploads/2022/01/Spectrum-Leasing-5G-Era.pdf

[10] *Traficom*. Accessed: Aug. 21, 2023. [Online]. Available: https://www.traficom.fi/fi/

[11] R. Wang, L. Liu, and W. Shi, "HydraSpace: Computational data storage for autonomous vehicles," in *Proc. IEEE 6th Int. Conf. Collaboration Internet Comput. (CIC)*, Dec. 2020, pp. 70–77, doi: 10.1109/CIC50333.2020.00033.

[12] A. Kostopoulos, I. P. Chochliouros, E. Sfakianakis, D. Munaretto, and C. Keuker, "Deploying a 5G architecture for crowd events," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6, doi: 10.1109/ICCW.2019.8757152.

[13] A. Felski and K. Zwolak, "The ocean-going autonomous ship—Challenges and threats," *J. Mar. Sci. Eng.*, vol. 8, no. 1, p. 41, Jan. 2020, doi: 10.3390/jmse8010041.

[14] M. Höyhtyä, J. Huusko, M. Kiviranta, K. Solberg, and J. Rokka, "Connectivity for autonomous ships: Architecture, use cases, and research challenges," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2017, pp. 345–350, doi: 10.1109/ICTC.2017.8191000.

[15] S. Yıldız, S. Kıvrak, G. Arslan, and Y. University, "Using drone technologies for construction project management: A narrative review," *J. Construct. Eng., Manage. Innov.*, vol. 4, no. 4, pp. 229–244, Dec. 2021, doi: 10.31462/jcemi.2021.04229244.

[16] I. Mesogiti, E. Theodoropoulou, K. Filis, G. Lyberopoulos, A. Ropodi, K. Tsagkaris, P. Demestichas, N. Pleros, G. Kalfas, and C. Vagionas, "Fiber-wireless fronthaul/backhaul network architectures for 5G," in *Proc. IEEE 23rd Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Sep. 2018, pp. 1–5, doi: 10.1109/CAMAD.2018.8514991.

[17] "Minimum requirements related to technical performance for IMT-2020 radio interface(s)," ITU Union, Geneva, Switzerland, Tech. Rep. 2410, Nov. 2017. [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2410-2017-PDF-E.pdf

[18] *Small Cell Solutions—Urban Network Connectivity—Ericsson*. Accessed: Nov. 11, 2023. [Online]. Available: https://www.ericsson.com/en/small-cells

[19] *Why the 5G Network Needs 'Small Cells' Technology*. Accessed: Aug. 21, 2023. [Online]. Available: https://www.linkedin.com/pulse/why-5g-network-needs-small-cells-technology-patrick-mutabazi?utm_source=share&utm_medium=guest_desktop&utm_campaign

[20] P. Ryan, "Treating the wireless spectrum as a natural resource," *Environ. Law Rep.*, vol. 35, p. 10620, Sep. 2005. [Online]. Available: https://ssrn.com/abstract=793526

[21] *5G; NR; Physical Channels and Modulation (3GPP TS 38.211 Version 15.2.0 Release 15)*, document TS 138 211-V15.2.0, ETSI, Jul. 2018. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/15.02.00_60/ts_138211v150200p.pdf

[22] *European Conference of Postal and Telecommunications Administrations*. Accessed: Aug. 20, 2023. [Online]. Available: https://www.cept.org/

[23] *European Telecommunications Standards Institute*. Accessed: Aug. 21, 2023. [Online]. Available: https://www.etsi.org/

[24] *Wireless Innovation Forum | CBRS, SDR & Spectrum Sharing Standards*. Accessed: Aug. 21, 2023. [Online]. Available: https://www.wirelessinnovation.org/

[25] *ONGO Wireless Coverage—In-Building, Public Space & Industrial IoT | CBRS Alliance*. Accessed: Aug. 21, 2023. [Online]. Available: https://ongoalliance.org/

[26] *Decision, no. 243/2012/EU of the European Parliament and of the Council*. Mar. 2012. Accessed: Aug. 21, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32012D0243&from=EN

[27] RSP Group. *RSPG21-016 Final RSPG Report*. Feb. 2021. Accessed: Aug. 21, 2023. [Online]. Available: https://rspg-spectrum.eu/wp-content/uploads/2021/02/RSPG21-016final_RSPG_Report_on_Spectrum_Sharing.pdf

[28] (2021). *What is CBRS? How it Works & Why Your Enterprise Should Care*. Accessed: Aug. 18, 2023. [Online]. Available: https://www.celona.io/cbrs/what-is-cbrs

[29] M. Massaro and F. Beltrán, "Will 5G lead to more spectrum sharing? Discussing recent developments of the LSA and the CBRS spectrum sharing frameworks," *Telecommun. Policy*, vol. 44, no. 7, Aug. 2020, Art. no. 101973. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308596120300653, doi: 10.1016/j.telpol.2020.101973.

[30] B. Bahrak, S. Bhattarai, A. Ullah, J. J. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DYSPAN)*, Apr. 2014, pp. 236–247, doi: 10.1109/DySPAN.2014.6817800.

[31] *Blockchain and Identity: Projects/Companies Working on Blockchain and Identity*. Accessed: Aug. 19, 2023. [Online]. Available: https://github.com/peacekeeper/blockchain-identity

[32] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele, and C. Allen. *Decentralized Identifiers (DIDs) V1.0*. Accessed: Aug. 20, 2023. [Online]. Available: https://www.w3.org/TR/did-core/

[33] *DIF—Decentralized Identity Foundation*. Accessed: Aug. 21, 2023. [Online]. Available: https://identity.foundation/

[34] A. Tobin and D. Reed, "The inevitable rise of self sovereign identity," Sovrin Found., USA, White Paper, Sep. 2016. [Online]. Available: https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf

[35] M. Sporny, D. Zagidulin, and D. Longley. (2022). *The Did: Key Method V0.7*. Accessed: May 9, 2023. [Online]. Available: https://w3c-ccg.github.io/did-method-key/

[36] N. Fotiou. (2022). DID: Self Method Specification. MMLab/AUEB, ExcID. Accessed: May 9, 2023. [Online]. Available: https://github.com/excid-io/did-self

[37] *Hyperledger Indy—Hyperledger Indy 1.0 Documentation*. Accessed: Aug. 21, 2023. [Online]. Available: https://indy.readthedocs.io/en/latest/

[38] *What is GDPR, the EU's New Data Protection Law?—GDPR.eu*. Accessed: Aug. 21, 2023. [Online]. Available: https://gdpr.eu/what-is-gdpr/

[39] M. Sporny, D. Longley, and D. Chadwick. *Verifiable Credentials Data Model V1.1*. Accessed: Aug. 19, 2023. [Online]. Available: https://www.w3.org/TR/vc-data-model

[40] K. Wüst and A. Gervais, "Do you need a blockchain?" in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 45–54, doi: 10.1109/CVCBT.2018.00011.

[41] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[42] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," ethereum.org, Switzerland, Tech. Rep., 2014. [Online]. Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf

[43] N. Fotiou and G. C. Polyzos, "Smart contracts for the Internet of Things: Opportunities and challenges," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2018, pp. 256–260, doi: 10.1109/EuCNC.2018.8443212.

[44] *Private Data—Hyperledger-Fabricdocs Main Documentation*. Accessed: Sep. 5, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/private-data/private-data.html

[45] D. B. Rawat and A. Alshaikhi, "Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Mar. 2018, pp. 332–336, doi: 10.1109/ICCNC.2018.8390344.

[46] Y. Xiao, S. Shi, W. Lou, C. Wang, X. Li, N. Zhang, Y. T. Hou, and J. H. Reed, "Decentralized spectrum access system: Vision, challenges, and a blockchain solution," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 220–228, Feb. 2022, doi: 10.1109/MWC.101.2100354.

[47] E. Di Pascale, J. McMenamy, I. Macaluso, and L. Doyle, "Smart contract SLAs for dense small-cell-as-a-service," Mar. 2017, *arXiv:1703.04502*.

[48] A. Kostopoulos, I. P. Chochliouros, I. Giannoulakis, A. Kourtis, and E. Kafetzakis, "Small cells-as-a-service in 5G networks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2018, pp. 1–5, doi: 10.1109/MWC.101.2100354.

[49] P. Gorla, V. Chamola, V. Hassija, and N. Ansari, "Blockchain based framework for modeling and evaluating 5G spectrum sharing," *IEEE Netw.*, vol. 35, no. 2, pp. 229–235, Mar. 2021, doi: 10.1109/MNET.011.2000469.

[50] J. Kim, H. Cha, and S.-L. Kim, "Spectrum leasing for micro-operators using blockchain networks," in *Proc. 13th EAI Int. Conf. CROWNCOM*, Ghent, Belgium, Jan. 2019, pp. 66–77, doi: 10.1007/978-3-030-05490-8_7.

[51] M. Liu, Q. Wu, Y. Hei, D. Li, and J. Hu, "Fair and smart spectrum allocation scheme for IIoT based on blockchain," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102686. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157087052100192X, doi: 10.1016/j.adhoc.2021.102686.

[52] Z. Tu, K. Zhu, C. Yi, and R. Wang, "Blockchain-based privacy-preserving dynamic spectrum sharing," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, Sep. 2020, pp. 444–456, doi: 10.1007/978-3-030-59016-1_37.

[53] *RFC 6749: The Oauth 2.0 Authorization Framework*. Accessed: Apr. 11, 2023. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6749

[54] *Fabric-Samples/Test-Network at Main—Github*. Accessed: Jan. 5, 2023. [Online]. Available: https://github.com/hyperledger/fabric-samples/tree/main/test-network

[55] *Tutorial: Get Started With Go—The Go Programming Language*. Accessed: Sep. 5, 2023. [Online]. Available: https://go.dev/doc/tutorial/getting-started

[56] *Apache CouchDB*. Accessed: Jan. 3, 2023. [Online]. Available: https://couchdb.apache.org/

[57] *Docker Documentation*. Accessed: Mar. 1, 2023. [Online]. Available: https://docs.docker.com/

[58] *Hyperledger Caliper | Caliper is a Blockchain Performance Benchmark Framework, Which Allows Users to Test Different Blockchain Solutions With Predefined Use Cases, and Get a Set of Performance Test Results*. Accessed: Sep. 5, 2023. [Online]. Available: https://hyperledger.github.io/caliper/

[59] *CBRS Priority Access License Auction Fact Sheet—Ongo Alliance*. Accessed: Jul. 3, 2023. [Online]. Available: https://ongoalliance.org/resource/cbrs-priority-access-license-auction-fact-sheet/

MUKESH THAKUR received the M.Sc. degree in computer science from the University of Helsinki, Finland, in 2017. He is currently pursuing the Ph.D. degree in computer science.

His research interests include mobile networks, distributed ledgers and blockchains, decentralized identifiers and verifiable credentials, and future network technologies.

YKI KORTESNIEMI received the M.Sc. (Tech.) degree in industrial management and the Lic.Sc. (Tech.) degree in computer science from the Helsinki University of Technology, Finland, in 1998 and 2003, respectively, and the D.Sc. (Tech.) degree in networking technology from Aalto University, Finland, in 2015.

He has worked on numerous research projects with the Helsinki University of Technology and Aalto University, including the EU H2020 Projects SOFIE and IoT-NGIN. His research interests include information security and privacy, data protection, MyData and legal design, the Internet of Things, distributed ledgers and blockchains, and decentralized identifiers and verifiable credentials.

DMITRIJ LAGUTIN received the M.Sc. (Tech.) degree from the Helsinki University of Technology, Finland, in 2005, and the D.Sc. (Tech.) degree from Aalto University, Finland, in 2010.

He has been a Researcher in several research projects with the Helsinki University of Technology and Aalto University, including EU FP7 PSIRP, PURSUIT, EU H2020 POINT, SOFIE, and IoT-NGIN Project with Aalto University. His research interests include network security and privacy, the Internet of Things, blockchains, and future network technologies.