Distributed ledger technologies for authentication and access control in networking applications: a comprehensive survey

Fariba Ghaffari, Emmanuel Bertin, Noel Crespi, Julien Hatin

Distributed Ledger Technologies for Authentication and Access Control in Networking Applications: A Comprehensive Survey

Fariba Ghaffari[a,b], Emmanuel Bertin[a,b], Noel Crespi[b], Julien Hatin[a]

[a]*Orange Telecom, France*
[b]*Institut Polytechnique de Paris, IMT, Telecom SudParis, France*

Abstract

The accelerated growth of networking technologies highlights the importance of Authentication and Access Control (AAC) as protection against associated attacks. Controlling access to resources, facilitating resource sharing, and managing user mobility are some of the notable capabilities provided by AAC methods. Centralized methods are the most common deployment architectures, that can be threatened by several attacks at their central points. Emerging Distributed Ledger Technology (DLT) has attracted significant interest in the AAA community. The distributed nature of DLT and its immutability can bring unprecedented opportunities to resolve many of the challenges of conventional systems. We survey the state-of-the-art in deploying authentication and access control approaches via DLT for several networking use cases. More precisely, we explore DLT applications in 1) Authentication; 2) Access Control; and 3) Comprehensive AAC solutions. First, we present the challenges of centralized solutions and discuss the capability of DLT for their resolution. Then, we propose a taxonomy to categorize the existing methods. Analysis, comparison, and discussion on the advantages and disadvantages of these methods have been provided regarding different parameters such as DLT types, AAC approaches, security, reliability, scalability, etc. While DLT provides various benefits, several challenges remain for the migration to DLT-based AAC. In light of these general limitations, we propose some future directions, targeting the current lacunae and future needs.

Keywords: Authentication, access control, networking applications, distributed ledger technology, Blockchain, smart contract, security, privacy, taxonomy.

1. Introduction

Due to the dramatic increase in the application of networking technologies, controlling access to resources is one of the vital challenges to be addressed. Authentication and Access Control (AAC) mechanisms play an undeniable role in resolving security and privacy problems. Authentication and access control complement each other in the process of providing legitimate access to a shared resource and strengthening network security [1]. Authentication is the act of verifying that the subjects (i.e., someone/something that wants to use a resource) are what they claim to be and that they are known by the system. Access control (authorization) is the process of accepting or denying the access request of an authenticated subject to a specific object (i.e., resources that the subject wants to use) [2].

With the increasing importance of AAC, various solutions have been proposed. A considerable part of the literature is dedicated to centralized systems. Despite the low complexity in the implementation of such methods and their high performance, they suffer from a single point of failure, the risk of privacy leakage in third-party agents, low scalability, high maintenance costs, and lack of audibility/transparency [3]. Any solution bringing high fault tolerance, integrity, non-repudiation, low maintenance cost, traceability, and permanency would be a solid candidate to change the future of AAC.

Distributed Ledger Technology (DLT) offers these unprecedented opportunities to potentially revolutionize AAC. Blockchain technology (as the first extension of DLT) [4] emerged in 2008 to support cryptocurrencies. In 2014, with the first implementation of smart contracts [5, 6], Ethereum made its first appearance. These technologies are changing many aspects of business models, management, and operations in the IoT [7, 8], smart cities [9, 10], cloud computing [11, 12], edge computing [13], fog computing [14], industry 4.0 [15], big data [16], etc. The AAA community also benefits from DLT in a variety of network technologies (i.e., communication networks, cloud computing, the IoT and smart cities, etc.).

In this article, we review the current DLT-based AAC methods for different networking applications. It is important to mention that, due to the higher maturity of Blockchain (rather than other DLT platforms), the majority of proposed methods use Blockchain and smart contracts. Based on our findings, we propose a taxonomy for classifying the existing approaches based on their characteristics (e.g., their approach to using DLT, the role of DLT in their solution, and DLT types). Moreover, we list their advantages and disadvantages concerning security capabilities, time consumption, cost-effectiveness, and performance. This information guides us to suggest several future directions.

A summary of ***paper organization*** is depicted in Fig. 1.

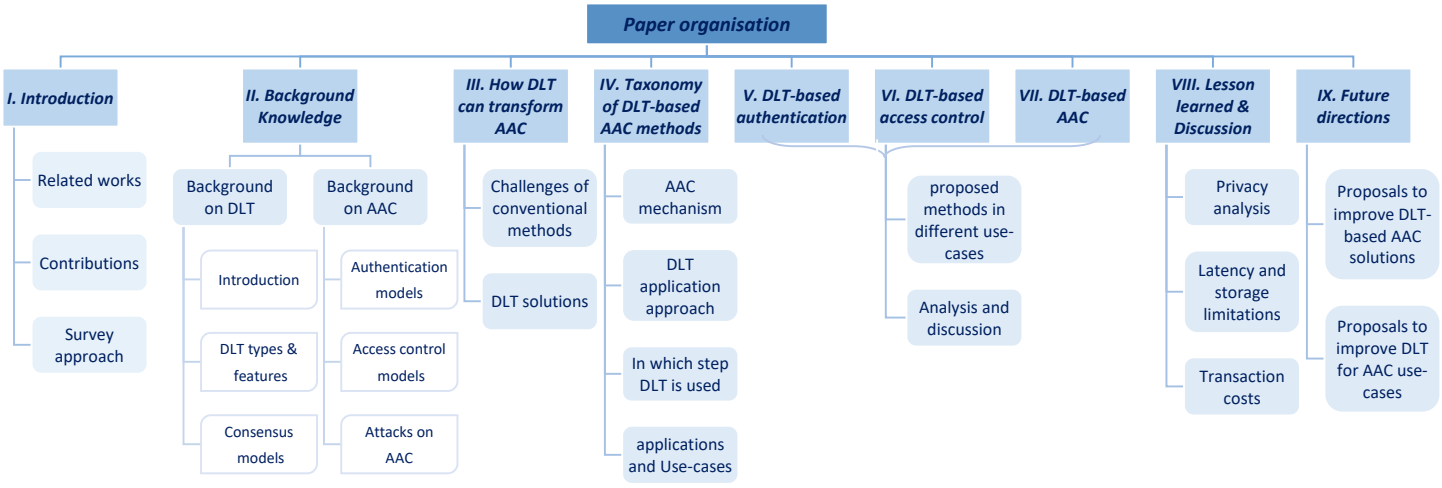

Figure 1: Organization of the paper

1.1. Related Works and contribution

Many efforts to benefit from the advantages of DLT have been proposed recently for diverse use cases, and several papers survey this technology from different perspectives [17, 18]. The works in [9, 19, 20, 21, 22, 23] review Blockchain applications in the IoT and smart cities. Gai et al. [12] is focused on Blockchain in cloud computing, while the cloud of things is investigated in [11]. Edge computing and Fog computing systems and their challenges in Blockchain are surveyed in [13, 24] and [14], respectively. Blockchain applications for 5G and beyond in various aspects of user/entity connections to the network, such as identity management, authentication, and network slicing are surveyed in [25, 26]. Moreover, Perez et al. [27] surveyed smart contract-based crowd-sourcing methods to improve security and privacy-preserving.

From the security perspective, Salman et al. [28] surveyed the Blockchain-based approaches for several security services. Although this paper is marginally similar to this work, there are several fundamental differences. For instance, [28] is focused on various Public Key Infrastructures (PKI) for authentication solutions based on Blockchain, while we target different authentication methods (rather than only PKI-based solutions). Moreover, the access control in [28] is briefly discussed about Blockchain-based access control lists (ACLs), while we comprehensively surveyed a variety of Blockchain-based access control methods. Furthermore, while we consider the application environment in which Blockchain is used and the role of Blockchain in the procedure, these types of analyses are not provided in [28]. Lim et al. [29, 30] investigated Blockchain's benefits in identity management and authentication. Our previous works [31, 32] briefly survey this area, with more limited analysis. Moreover, [33, 34, 35] surveyed the Blockchain in information systems management, privacy, and security.

Although DLT and its applications have been covered by many surveys (some of them are mentioned previously), our study offers the following unique contributions:

1. To the best of our knowledge, this is the first work exploring DLT usage, particularly in authentication and access control for networking technologies and applications;
2. It offers a systematic and taxonomic approach to surveying the state of the art to better categorize the methods;
3. The challenges in using DLT for AAC methods are studied; moreover, the efficiency and limitations of the existing solutions are discussed. The outputs of the analyses are our support for suggesting several directions for future works in the AAC and DLT communities;
4. This survey's analysis of DLT and its impact on AAC mechanisms in terms of security, privacy, performance, etc., can help researchers to determine the best solutions for their future projects; and
5. The proposed future directions are considered separately for two evolved communities (DLT and AAC). This separation can help each community to better determine its role in this process.

We compared our survey with other related works in Table 1. In this table, we only compared the relevant studies in which, at least, an overview of authentication or access control using DLT is provided. As listed in the table, our paper brings more advantages regarding the analysis of the proposed method in terms of different criteria and the focus on the existing methods in the networking field. Moreover, we studied the state of the arts more comprehensively from 2014 to 2022.

1.2. Survey approach

In this paper, we undertake a Systematic Literature Review (SLR) approach [36, 37] to analyze the existing DLT-based AAC

Table 1: Comparison among existing surveys related to AAC methods based on DLT

Params		***[28]***	***[29]***	***[31, 32]***	***[33]***	***[21]***	***[35]***	***[25]***	***This Survey***
Targeted application domains for Authentication		PKI-based methods in IoT and decentralized sensor networks.	Commercial authentication methods	Cellular networks, IoT, healthcare, and cloud computing	Authentication in information systems	IoT use-cases	Authentication methods for IoT use-cases	Authentication methods in 5G (two specific examples)	***IoT, Telecom, Smart healthcare, WSNs, ICN, and Cloud***
Targeted application domains for access control		ACL-based methods in IoT and cloud	Access control methods *are not surveyed*	Cellular networks, IoT, healthcare, and cloud	Access control in information systems	IoT use-cases	IoT use-cases	Access control methods *are not surveyed*	***IoT, Telecomm, Smart healthcare, ICN, and Cloud computing***
Indicating AAC use cases in application domain++		No	No	No	No	No	No	No	***Yes***
Security	***Privacy analysis***	Yes	No	No	No	No	Analysis	No	***Analysis****
	Security analysis	No	No	No	No	No	No	Limited	***Yes***
Method analysis	***AAC type***	No	No	No	No	No	No	No	***Yes***
	DLT approach*	No	No	No	No	No	No	No	***Yes***
	AAC step*	No	No	No	No	No	No	No	***Yes***
	Use-case*	No	No	Yes	No	No	No	Yes (5G)	***Yes***
	AAC purpose*	No	No	No	No	No	No	No	***Yes***
	DLT network*	Yes	Yes	No	No	No	No	No	***Yes***
	DLT type*	No	Yes	No	No	No	No	Yes	***Yes***
Analyzing pros and cons of the surveyed methods		No	No	No	No	No	Yes	No	***Yes***
Analyzing DLT opportunities for existing AAC		No	No	No	No	Yes	No	Yes	***Yes***
Comparison among surveyed DLT-based AAC methods		Yes	No	No	No	No	No	Yes	***Yes***
Taxonomic review		No	No	Yes	No	No	No	No	***Yes***
Related to networking field		Yes	No	No	No	No	No	Yes	***Yes***
Providing future direction		Yes+	No	No	No for AAC	No	Yes	Yes	***Yes***
Future direction for AAC and DLT communities		No	No	No	No	No	No	No	***Yes***

* In this paper, we analyzed the impact of using DLT for AAC methods on privacy. It is important to mention that the analysis of the existing DLT-based privacy-preserving methods is not in the scope of this paper.

** The principal descriptions of these criteria and their definitions are provided in Section 4.

+ In this paper the future directions are not directly discussed, but the challenges of Blockchain technology are provided.

++ For instance in an IoT network, access control use-cases can be the right delegation, data sharing, access to sensors, and network security (See Fig. 4)

methods and to form a better perception of using DLT in AAC procedures in different use cases. SLR approach is a method to identify and evaluate the state-of-the-art in a specific field using a particular theme [38]. In the SLR process, we answer the following questions in every step [38]:

1. *Research questions:* In this step we need to clearly set the questions which need to be addressed at the end of the study.
2. *Research process:* In this step we need to clearly define the process of our research including the name of utilized research databases, journals, conferences, subjects, etc.
3. *Inclusion and exclusion criteria:* In this step we need to clearly define which type/subjects of retrieved papers in the previous step are needed to be included in our investigation, and which of them should be excluded. The selection can be based on publication type, publication year, research area, etc.
4. *Quality assessment:* In this step, we need to define several criteria to assess the quality of previously founded research and select several works based on these criteria.
5. *Data collection:* In this step we need to clearly define which type of data is needed to be extracted from the selected works.
6. *Data analysis:* In this step we need to clearly define how to present data in the survey. Moreover, we define that each information presentation aims to address which question.

In summary, the SLR procedure is done as follows. Note that, the detailed procedure is depicted in Fig. 2. First, we performed a systematic literature review by searching all the papers dealing with *comprehensive AAC solutions* (both authentication and access control as one method) that relied on DLT published in journals from 2008 to 2022. This step resulted in a relatively low number of papers. We then extended our search by including DLT-based authentication *or* access control solutions published in journals, conferences, or as theses in universities. This search resulted in a large number of methods in different use cases for a variety of purposes (i.e., more than 200 papers). Next, we filtered the papers that were related to authentication and access control in networking use cases (e.g., some of them only provide a solution for accessing a central database or physical access to an environment). Finally, after screening the papers, abstracts, and keywords for general relevance, applications/use cases, and relations to the networking and communication field, we selected almost 100 articles.

The selected papers (i.e., 98 papers) were read and systematically analyzed based on several parameters of AAC systems (i.e., AAC type, use cases, motivation to use DLT, and the step of AAC in which DLT is used). It is important to mention that a majority of the DLT-based AAC methods studied have been implemented based on Blockchain and smart contracts (compared to other extensions of DLT).

2. Background knowledge

2.1. Background on Distributed Ledger Technology

DLT is a general term for technologies that utilize replicated, shared, and synchronized digital data among the users of private or public distributed computers located on multiple sites[39]. Immutability, distributed/decentralized nature, consensus, transparency, non-repudiation, and being append-only are the common feature of all DLTs. Any change in the state or the value in the ledger can be accomplished through consensus among the nodes. Increasing the number of nodes participating in the consensus procedure decreases the probability of monopolization of the network by several malicious nodes. Also, with more extracted blocks, the immutability of the information is improved [39].

DLT-based platforms can be divided into two main categories, based on their deployment and access permissions [18]. ***Permission-less (Public)*** platforms are accessible to the public, and anyone can participate in consensus, read the transactions, and write in the ledger. All of the transaction records are available to all users. ***Permissioned*** platforms can be divided into two subcategories. *Private platform* is developed in an organization based on their needs; *Consortium platform* can be used as a distributed and reliable database for pre-defined enterprises for business-to-business purposes. In permissioned DLTs, only the eligible nodes, defined by participated organization(s), can join in the consensus process. So, the user's anonymity can be violated. Moreover, the tokens or fees are not mandatory for the process or validation of transactions.

DLT can be grouped into different categories based on their data structure [40]. For instance, the three following types of DLTs are the most well-known variations:

- *Blockchain*: In 2008, this technology was introduced to support cryptocurrencies in the financial sector. After the introduction of smart contracts [6] in 2014 [5], several applications such as stocks, loans, mortgages, and smart property were added to Blockchain. The main objectives of smart contracts are to satisfy common contractual conditions, minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Blockchain is a distributed ledger, structured into a linked list of blocks that contain an ordered set of transactions. To create a link with the previous block, each block uses the hash of the previous block. The number of transactions in each block can be varied based on the number of input transactions per second and the difficulty of the consensus puzzle. In its structure, each block has a header and a body. Most of the block headers have the following parameters: 1) a block version; 2) the hash of the previous block; 3) a hash of the Merkle tree root that stores the hash amounts of all transactions in the current block; 4) a timestamp for traceability; 5) a random number as a nonce; 6) the hash amount of all the data in the header and body of the current block.
- *BlockDAG*: Block Directed Acyclic Graph (BlockDAG) replaced the linked-list structure of Blockchain with the

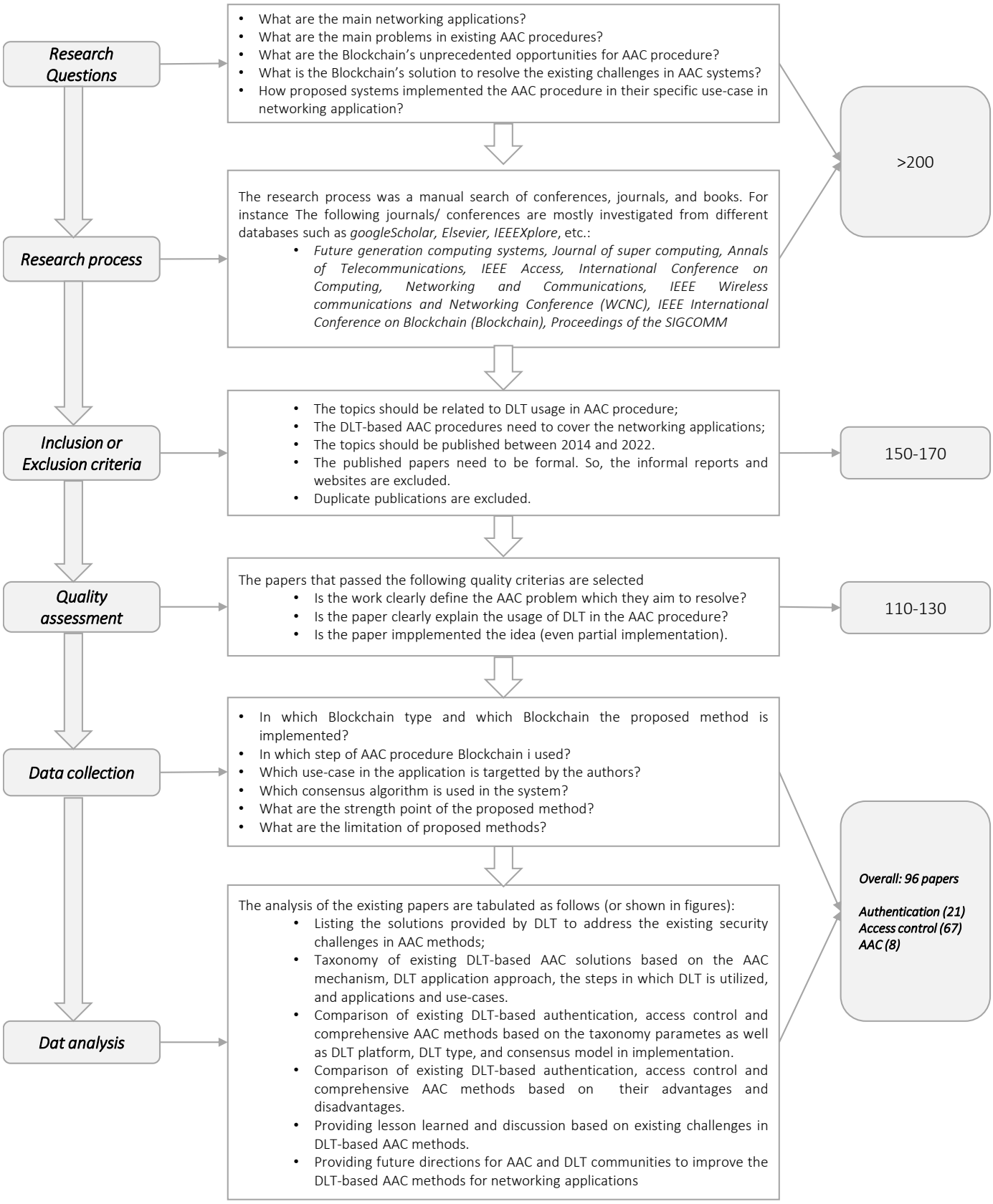

Figure 2: Survey approach: Systematic Literature Review

DAG [41]. The main hypothesis of BlockDAG is to serve/ validate transactions and blocks as fast as possible. To provide consistency in the system, the miners of new blocks decide on the order of the transactions [42]. Tangle is an example of BlockDAG [43].

- *TDAG*: Transaction-based DAG or Block-less DAG removes the concept of the block. The impetus for this technology is that even in BlockDAG, different blocks may contain overlapped transactions which can increase the bandwidth requirements. So, in TDAG, transactions are linked directly together in the DAG structure, and there are no blocks at all [40]. IOTA and Nano are two examples of TDAGs.

2.1.1. DLT features

All of the above-mentioned DLT-based platforms share the following common characteristics:

- *Immutability*: it means that no confirmed transaction or data in DLT can be altered. Thanks to using the hash of the preceding block, any simple modification in a transaction/block requires solving a consensus problem for all of the subsequent blocks.
- *Decentralized and fault tolerant*: it means that there is no central authority to control the network, and failure of one or several nodes cannot harm the systems' functionality. So, there is no single point of failure in the DLT-based systems, and they provide high fault tolerance.
- *Reaching consensus*: all nodes in a DLT can reach a consensus, based on algorithms defined to ensure that all nodes have the latest version of the ledger. Thanks to this characteristic, the integrity of data and transactions is maintained.
- *Traceability/ Transparency*: it means, in the distributed ledger, all transactions are available to be seen and tracked by the nodes. So, all data is always available and traceable at any time. This feature can be especially useful in forensics [44, 45].
- *Non-repudiation*: it means no one can deny their actions in the DLT-based networks. Thanks to using the users' private-key-based signatures on each transaction, the possibility of action denial can be eliminated.
- *Permanence*: this feature means that all data in a DLT can be available at any time (nothing may be removed from the network).

2.1.2. Consensus Mechanisms

A consensus mechanism is a sequence of steps followed by all or most of the nodes in a DLT-based system to reach an agreement on a proposed state or value. The validity, agreement, termination, and fault tolerance are the most important requirements of consensus mechanisms [46, 39, 47]. The consensus methods can be categorized into three different groups [48]: 1) compute-intensive based, 2) capability-based, and 3) voting based. In the following subsections, we briefly introduce the well-known methods used in the studied AAC solutions.

Compute-intensive based consensus algorithms: These algorithms require a substantial amount of computing resources to solve the consensus problem. One of the most well-known examples of this type is Proof-of-Work (PoW)[49, 50] method. PoW algorithm works based on the framework of a cryptographic block-discovery racing game. Nodes (known as miners) try to solve a mathematically complex puzzle that uses a tremendous amount of their computational resources. The first miner, that finds the result, is the winner who can broadcast the result to all the nodes in the network (i.e., via the gossiping rule). Bitcoin deployed the PoW protocol [4].

Capability-based consensus algorithms: Due to the energy inefficiency of compute-intensive-based consensus algorithms, other alternatives have been proposed. Capability-based algorithms rely on the capabilities of nodes instead of their computational power. *Proof of Stake (PoS)*[51] is the most well-known algorithm in this category. In PoS, the block validator (the only responsible node for generating the next block) is selected based on the stakes it would have(i.e., coins or tokens owned by a node). *Proof of Authority (PoA)*[52] is a reputation-based method in which the reputation of the validator is the capability parameter. The validators (authorities) in PoA have formally approved accounts, and their identity is kept public [53].

Voting-based consensus algorithms: Implementing technological democracy, in the voting-based consensus algorithms, the miners and validators are selected based on the voting process among network nodes. For instance, *Delegated Proof of Stake (DPoS)*[54] relies on selecting delegates (witnesses) instead of the validators of the blocks. The witnesses can be interpreted as trusted nodes in the network, chosen by an election, to validate the blocks instead of nodes. Another example in this category is *Practical Byzantine Fault Tolerance (PBFT)* [55] in which there is one "leader" node and several "backup" nodes. This method has five steps named "Request", "Pre-prepare", "Prepare", "Commit" and "Reply". This method is energy efficient, but its scalability is limited [39]. Moreover, *Raft Algorithm*[56] is a consensus method in which, at any time, every node is in one of the three states as *leader, follower*, or *candidate*. The leader serves the network until it crashes. When a leader fails the election process starts to select another "leader" from the "candidate" list. Then the "candidate" requests votes from other nodes to become a "leader". *Ouroboros* [57] is another algorithm in this category that is based on PoS. In this method, time is divided into fixed-time epochs. In each epoch, the electors can be selected based on the weight of the stake of the stakeholder.

The pre-mentioned consensus models are compared in Table 2, based on the following parameters [39, 58, 59]:

- *Byzantine Failure Tolerance (BFT):* The maximum tolerable rate of Byzantine nodes in the system.
- *Scalability:* The system's ability to tolerate the increasing number of nodes.

- *Throughput/ transaction rate:* The average number of transactions validated in one second.
- *Recourse consumption:* The number of resources needed for the operation of the method.
- *Recourse type:* All types of resources needed to run the method by a specific node (e.g., computational power, reputation, stake).

2.2. Background on AAC mechanisms

This section provides an overview of Authentication and Access Control (AAC) methods. As mentioned earlier, authentication and access control complete each other to build the foundation of operational networks and applications by providing secure access to network resources or data.

2.2.1. Overview of Authentication

Authentication is a security mechanism that verifies who is the client sending the request and that they are the users they claim to be [2]. This process is accomplished through the following steps: 1) The eligible credentials or identities would be stored in an authentication server; 2) A registered user sends a request by providing the required data; 3) The authentication server records the complete log of the connection request; 4) The authentication server compares the received data with the stored identification in a database (verification); and 4) If the data matches, the verification is successful, and the user can log into the system (e.g., by providing a login solution). Authentication procedure can be implemented in different ways;

- *Knowledge-based* methods rely on the users' knowledge about specific questions, such as identities (IDs), passwords, PIN codes, etc.
- *Possession-based* methods operate based on something that the user possesses; For instance, Radio Frequency Identification (RFID) card.
- *Biometric-based* methods rely on one or more physical features of the user such as fingerprints. These methods are also termed Inherence-based authentication [61].
- *Multi-factor authentication* methods combine two or more different solutions to make the authentication more secure. For example, a user may enter her password and a security code sent by SMS to her phone.

2.2.2. Overview of access control methods

Access control regulates who or what (i.e., subject) can perform which action (or have which permissions) on an object (e.g., network resource, database) [62]. The access control procedure is done in three main steps: 1) Policy/ rule definition that determines the rules of accessing an object. Each rule definition is varied based on the access control model. 2) Access verification, in which the access control server examines the received access request based on a subject's permissions. If they match, an access solution based on the enforcement method will be assigned to the subject. 3) Recording access logs, in which all activities of the subjects and their accesses will be recorded.

In this section, we describe the well-known access control methods implemented in the investigated articles. Note that, due to the large quantity of access control models, presenting and analyzing all of them is out of the scope of this paper.

In *Discretionary Access Control(DAC)* [63] considers the owner-based administration of objects. More precisely, the owner of an object defines the access rules and policies. DAC can be implemented via an Access Control List (ACL) that defines which objects can be accessed by which subject with what type of permission. A similar access control method is *Capability-based Access Control(CapBAC)*[64] in which a capability is associated with each subject and used for access management. In the CapBAC model, users are granted access permissions based on an access token, such as a key, a ticket, a credential, etc. [65]. When a system aims to manage a large number of assets, CapBAC and DAC decrease the manageability [66]. So, *Role-Based Access Control (RBAC)* is developed to resolve this challenge. It manages the subjects' access, based on their role within the system, and also defines what kind of accesses are associated with the subject of a given role [67].

Moreover, *Attribute-Based Access Control (ABAC)* [68] is a logical model that controls access to objects by evaluating some defined access control rules or policies in terms of the "subject, object, action" and "environment" attributes, that specify the subject, object, allowed operations of the subject on an object, and the context in which the access is requested, respectively. *Attribute-based Encryption (ABE)* is a novel model of providing attribute-based access control while preserving data confidentiality [69, 70]. ABE encrypts data without any exact knowledge of the receiver. A user's secret key and ciphertexts are dependent upon some attributes. Ciphertext Policy ABE (CP-ABE) [71] is a popular variant of the ABE method, in which a user's secret key is associated with a set of attributes, and a ciphertext specifies an access policy. Data decryption will only be possible if the user's secret key satisfies the access structure with the associated ciphertext.

2.2.3. Main security attacks on AAC

Several types of attacks can target AAC procedures. Some of the well-known attacks are described below [72, 73, 74, 75]:

- *Password cracking:* Attackers try to find the identifications of legitimate users by recovering them from storage. The most well-known attacks in this category are brute-force (checks all possible answers), rainbow (generates the password hash table in advance), and dictionary (uses a sample dataset of the most-used passwords).
- *Denial-of-Service (DoS)/ Distributed DoS (DDoS):* The purpose of these attacks is to make a resource unavailable for legitimate users. Request flooding, ping of death, and SYN flood are well-known DoS/DDoS attacks.
- *Man-in-the-Middle (MitM):* The attacker relays information on behalf of the connection between source and des-

Table 2: Comparison of Existing Consensus Models

Type	*Feature Method*	*BFT*	*Scalability*	*Throughput*	*Resource consumption*	*Resource type*	*Example*
Compute-intensive	PoW	49%($N/2$)	High	low	High	Computational power	Bitcoin, Ethereum
Capability-based	PoS	49%($N/2$)	High	Low	Moderate	Stake	Ethereum, peerCoin
	PoA [52]	49%($N/2$)	High	Moderate	Low	Stake/ Reputation	VeChain
Voting-based	DPoS	49%($N/2$)	High	Moderate	Moderate	Reputation	EOS
	Raft [60]	49%($N/2$)	Low	Higher than PBFT	Moderate	Time	Quorum
	Ouroboros	33%($N/3$)	High	-	Low	Stake	Cardano
	PBFT	33% ($N/3$)	Low	High	Low	None	Hyperledger Fabric

tination, without their knowledge, and can alter, modify, or eavesdrop on their data. Another form of this attack is the *reply* attack, in which the attacker stores the user's identity data and uses that for subsequent connections.

- *Sybil:* In this type of attack, the attacker will define multiple virtual identities to target a network [76]. It means a single malicious node manages to influence the whole system using different identities.
- *Spoofing:* in this type of attack, the attackers impersonate another identity in the system, aiming to steal data, accelerate their privilege, or launch other malicious activities.

3. How DLT can transform AAC

To see how the AAC systems can be transformed by the emergence of the DLT, we need to identify the weak points of the existing solutions and the advantages of DLT at that point. So, the disadvantages of the existing AAC systems, and DLT's solution to these challenges are listed below [28, 29]:

- *Single point of failure:* Because of its centralized nature, the existing AAC systems suffer from a single point of failure. So, a crash in the centralized point can extremely affect the performance of the system. Moreover, it is more possible to compromise the central database. Thanks to *distributed nature* of the DLT, there is no need to have a central authority or database in the DLT-based systems.
- *Compromise of scalability and data integrity:* Having a centralized database brings the highest level of data integrity. In contrast, it suffers from scalability. Owing to *consensus* process in DLT, after reaching consensus in the system, all nodes have the same ledger and the same order of transactions. Moreover, because of its *distributed nature*, DLT is highly scalable. So, reconciling the data integrity and scalability is provided in such systems.
- *Low-auditability:* Low accountability and auditability are other security challenges in conventional systems. Using DLT, because all the transactions are validated and recorded with a timestamp, it is possible to verify and trace the previous transactions and logs [18]. So, *traceability* is provided. Moreover, due to its *non-repudiation*, the users' signature is required at each transaction. Therefore, no one can deny their action. Furthermore, based on the *immutability* of the DLT, no one can change the access or authentication logs or certificates in the system.
- *Data loss:* Centralized storage and server are highly vulnerable to data loss. Because vanishing the data in one storage/server results in losing all data without backup. Thanks to *permanency* of DLT, the final state of data is always available in the ledger of all nodes. This feature optimizes the advantages of *distributed* nature and *immutability*.

Besides its benefits for general challenges in conventional systems, DLT can improve the security of AAC regarding existing attack vectors. Table 3 lists the existing solutions in conventional systems to mitigate the attacks targeting AAC methods, their disadvantages, and the DLT-based solutions for them. Note that, despite the benefits of using DLT for AAC, there are some challenges, including 1) limited transaction processing capacity, 2) the lack of scalability in memory and storage, 3) the lack of knowledge about its robustness against different attacks, and 4) user privacy issues [77]. We will discuss these problems in Section 8.

4. Taxonomy of DLT-based AAC methods

In this section, we provide a taxonomy of existing DLT-based authentication, access control, and comprehensive AAC solutions using the SLR procedure explained in Section 1.2. Exploiting DLT in the AAC procedures of different use cases can influence various technologies, based on their specific needs. First provide the hierarchical architecture of DLT-based AAC solutions (see Fig. 3) to identify different technologies and their specific use-cases for AAC procedure.

Table 3: DLT solution for the main attacks on AAC

Attack	*conventional solutions*	*Disadvantages of conventional solution*	*DLT-based solution(s)*
Password cracking	Using password-less/multi-factor authentication	The user identifiers are stored in a central database and managed by a central authority that is vulnerable to attacks [75, 73, 78].	I) Password-less authentication along with distributed database for certifications eliminates the security problems of a central database [75]. II) Microsoft's ION [79] and Bitcoin's Identity protocol [78, 80], aim to provide secure identifier. III) Self-sovereign identity [81] is an alternative to central model of identity management.
	Account locking, after several wrong login attempts.	It may result in the locking of a legitimate user's account by the attacker.	
	Delayed response of the server to slow down the attacker	In large systems will result in high latency.	
	Strong password selection.	Not user-friendly.	
DoS/ DDoS	Using firewalls, IDSs, IPS, etc. to separate normal traffic and learn from attacks to avoid a similar pattern.	Can be ineffective because of the growing complexity and novelty of attacks.	I) The distributed nature of DLT can remove the single point of failure [75, 78]. II) The limited request generation rate in DLT makes DoS/DDoS attacks ineffective [82] in the application layer. Indeed it requires to limit the block size [72].
	Using redundant services to minimize the impact.	This solution changes the centralized architecture to a decentralized one.	
MitM/ Reply	Using SSL/TLS connections	SSL/TLS assumption is the trustworthiness of the central authority that issued the server key. If this assumption isn't satisfied the user may see a warning, and if they ignore it, MitM attacks are possible [86].	I) The user's signature on transactions and the block time-stamp [83, 84]. II) Owing to DLT's immutability, certificates can not be altered [87, 88]. III) To mitigate reply attacks, the secure identifiers can omit the session keys.
	Mutual authentication [73]		
	Adding timestamp and nonce to packets [85]		
Spoofing	Multi-factor authentication	The same problem of the central database in password cracking attacks.	I) The immutability of blocks in DLT, leads to assuring genuine user identity; II) The user's signature on transactions inoculate the system against spoofing attack [89].
	mutual authentication	The same vulnerability of certificate trustworthiness (see MitM attack)	
Sybil	using trusted certificates	Depends on the trustworthiness of a central authority (same as MitM attack)	I) To influence the whole system, the minimum number of adversary nodes must be more than the Byzantine fault tolerance, which makes the attack more complex [91]. II) The blocks (containing connection logs) are traceable [92]. So, an abnormal increase in the size of a chain can indicate the attack.
	Resource testing to ensure that the resources are matched with the number of unique identities [90].	Is not a solution to eliminating these attacks (it is a detection solution); Some studies show this method is ineffective [76].	

The architecture consists of six layers. The lowest layer is the *data layer*, which encapsulates the underlying block/transaction structure (e.g., linked list, DAG, etc.). Above that, the *network layer* contains the mechanisms of distributed networking, data propagation, communication among nodes, and data verification based on pre-defined structures (e.g., transaction verification via digital signature based on asymmetric cryptography). Next, the *consensus layer* mainly focuses on the consensus protocols of the nodes in the network (e.g., PoW, PBFT, PoS). These algorithms can have an incentive mechanism to encourage the nodes to collaborate in the consensus procedure and improve the security of the system. The *contract layer* coordinates the solutions' functioning based on smart contracts. This layer brings programmability into DLT. The two top layers in the architecture, *Authentication and Access Control* and *Application*, are related to the application of DLT in the desired context or use case (see Section 4.4). The authentication and access control layer aims to implement different AAC solutions for a variety of use cases (e.g., IoT, cloud, telecommunication, etc.) using DLT.

Based on the aforementioned SLR procedure, explained in Section 1.2, we categorized the existing authentication, access control, and comprehensive AAC mechanisms based on the four features shown in Fig. 4 and explained in the following subsections. It is important to mention that, Some studies worked on authentication and access control as one complete access-granting mechanism. These studies are discussed in Section 7 and are categorized based on all the mentioned features.

4.1. AAC mechanism

This feature defines the authentication or access control method implemented in the studied work. As mentioned in Section 2.2.1, Authentication types include Knowledge-based, Possession-based, Biometric-based, and Multi-factor solutions. Moreover, the studied access control methods cover DAC, CapBAC, RBAC, ABAC, and ABE-based solutions that are introduced in Section 2.2.2.

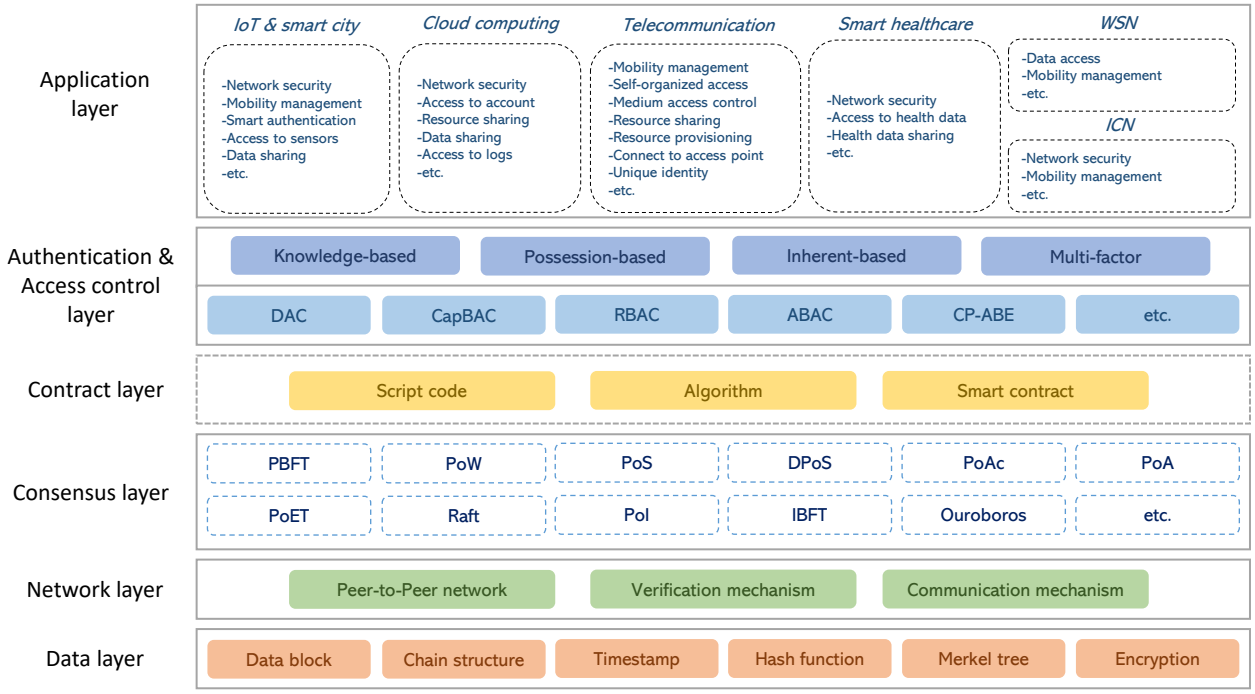

Figure 3: DLT-based AAC methods in networking applications [9, 93]

4.2. DLT application approach

According to our studies, we identified two general approaches for using DLT in authentication or access control procedures:

- Several studies use DLT as a distributed *database* to store credentials, identities, rules, roles, policies, and access logs. The main motivations of authors in these methods are the immutability, integrity, and permanence of DLT.
- In the considerable portion of literature, the authors use DLT not only as a secure database but also as a *decision point* for AAC procedure (e.g., to manage the authentication process by creating and handling the tokens, to handle the client's access based on predefined policies, enforcing the access decision, and storing the access log). Note that, in rare cases, the authors used DLT only as a decision point, not a database. Generally, distributed nature of DLT, removing the single point of failure, non-repudiation, permanence, and having programmable contracts, are the main motivations of the authors in these works.

4.3. In which step DLT is used

The authentication procedure can be done in four main steps: 1) log request to the system, 2) verification of the identity, 3) providing login solution, and 4) recording the access logs. In our study, we found out that rather than using DLT as a distributed *database for the credentials* and identities, it can be used in three steps out of the four aforementioned steps. It means the DLT (and more specifically smart contracts) can be used for *verification* of the user's identity to access the system, to *issue a one-time token* for the user's access, and to store the logs (i.e., *log management*).

The access control procedure can be executed in three main steps: 1) rule/ policy definition, 2) access verification, and 3) recording of the access logs. In our study about DLT-based access control solutions we concluded that the existing methods are using this technology for different purposes: 1) *defining the access policies* and rules in smart contracts, 2) *storing the access rules/policies* in smart contracts or Blockchain as a tamper-proof solution, 3) *verification* of the user's access request, 4) *enforcement* of access control decision, and 5) *recording the access logs*.

4.4. applications and Use-cases

Generally, enhancing system security against unauthorized access to data or resources is crucially dependent on AAC. However, in certain contexts, there are additional specific applications. For instance, in WSNs and IoT, the AAC procedure would be required in the mobility management of the nodes, while in cloud computing, it would accomplish the goal of sharing the resources. So, the following networking applications and their related use cases were identified for AAC methods. Note that, this section aims to expand the "Application layer" of Fig. 3.

- ***Internet of Things (IoT):*** The IoT refers to the interconnection among many context-aware products designed to collect, process, and communicate the data to make intelligent decisions [94]. IoT devices contain critical data for different environments such as smart cities, smart homes, wearable, and Wireless Sensor Networks (WSN). Access to these data requires secure solutions to preserve privacy

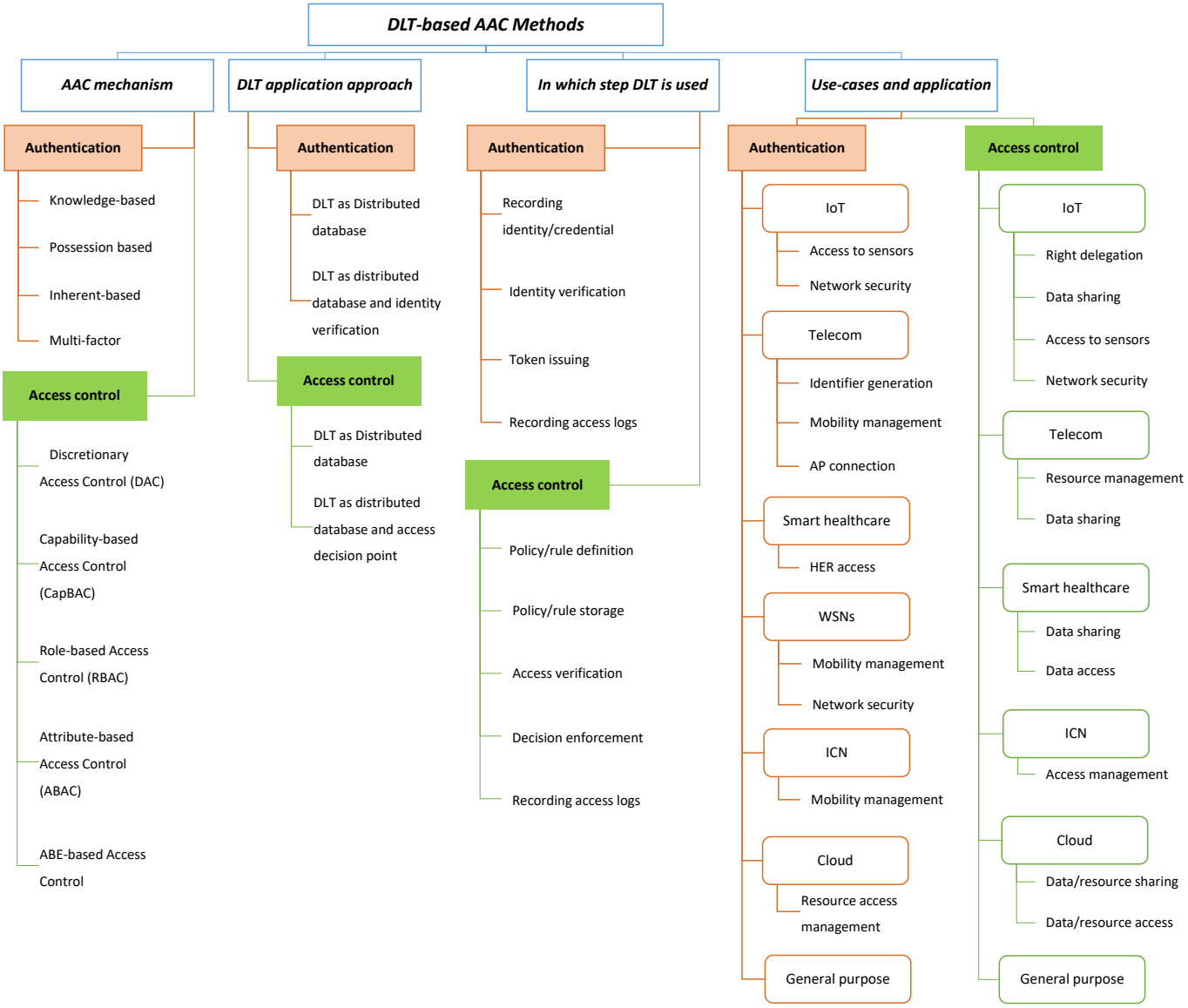

Figure 4: Taxonomy of existing AAC methods based on DLT.

and assure security. The applications of DLT-based AAC in the IoT include (but are not limited to): 1) network security; 2) mobility management of nodes in WSNs through different clusters; 3) providing secure access to the sensor data in smart homes/ cities; 4) the right delegation.

- ***Cloud computing:*** Cloud computing is a model for enabling ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources [95]. DLT and cloud computing are new advanced technologies that have a high potential for strengthening performance, security, and privacy in current web-based applications [12]. DLT-based AAC in a cloud environment has been targeted by several researchers as a means to 1) improve network security; 2) share the resources of the cloud computing environment, such as computing power and memory; 3) access the resource-sharing logs; and 4) facilitate data sharing in the cloud environment.
- ***Cellular networks and Telecommunication:*** Along with other technologies, telecommunication, and cellular networks can benefit from the advantages of DLT-based authentication and access control. Recent studies seek to deliver the following services in using DLT for AAC: 1) mobility management among different service and network providers; 2) provide self-organized access to the network; 3) enable medium access control by replacing new solutions with other existing methods such as Aloha [96]; 4) network resource sharing; 5) provide DLT-based user connections to the Wi-Fi access points instead of using knowledge-based authentication; and 6) generate DLT-based unique identities for users.
- ***Smart Healthcare:*** Smart Healthcare is involved with all type of technologies (e.g., IoT sensors) that leads to bet-

ter diagnosis of disease and sufficient treatment for patients. One of the challenging parts of this approach is to manage the electronic health records of the patients in a secure manner. So, several pieces of research proposed DLT-based AAC solutions to 1) provide overall security in the network to store patients' data, 2) share the patient's health records, with proper doctors, health agencies, and research departments along with preserve their privacy, and 3) manage access to the patient's records.

- ***Information Centric Networking (ICN):*** Information-Centric Networking (ICN) is a connection-less pull-based communication model that aims to distribute content in a highly scalable and efficient way via named data objects, such as web pages, videos, and documents [97]. Based on our research, the existing DLT-based AAC models in ICN are not only focused on protecting network security but also target producer mobility management (i.e., the challenging part is that ICN focuses on the named-based resolution mechanism) [98].

5. DLT-based Authentication Methods

This section presents the current authentication methods that rely on DLT (mostly Blockchain and smart contracts). Firstly, these methods are divided based on their application use cases, then the purpose of authentication, and finally, their approach to using DLT. In the final part of this subsection, a general discussion about the DLT-based authentication methods is provided.

5.1. Proposed methods for IoT/ smart cities

Access to IoT sensors

DLT as a distributed Database Huh et al. [99] proposed an automatic door-locking system via fingerprint-based authentication. The hash of a user's fingerprint is stored on the Blockchain and the users can authenticate themselves through mobile devices. In this method, the mobile phone executes a PoW consensus mechanism, a very resource-consuming task. Wu et al. [100] proposed a two-factor authentication method that uses an out-of-band channel to perform secondary authentication. The procedure begins with the Authentication Subject (AS) sending its requests to the object. The object retrieves the data of the AS from the Blockchain and performs a mutual authentication before revealing the data to AS.

DLT as a distributed database and verification solution: Ourad et al. [87] proposed a smart contract-based solution, in which the Ethereum address is the authentication identifier. The smart contract broadcasts an access token to the sender's Ethereum address if validation of the user is successful. The user combines and signs several data and sends them to the IoT device to verify. Having correct data, the device grants access to the user for the specified duration. This method runs the PoW consensus method which results in high resource consumption.

Network security

DLT as a distributed database and verification solution: Hammi et al. [101] proposed "bubbles of trust" in which each node will have a ticket based on its ID (i.e., Blockchain address). This approach creates secure virtual zones (bubbles) where IoT nodes can identify and trust each other. Time and cost efficiency are the main challenges of this method.

5.2. Proposed methods for telecommunication

Identifier generation

DLT as a distributed Database: Lee et al. [102] proposed BIDaaS that generates a Blockchain-based ID for users (instead of conventional IDs in cellular networks), and then this ID is registered on the Blockchain for further mutual authentication process. Protecting user privacy is the main challenge of this method.

Mobility management

DLT as a distributed Database: Yazdinejad et al. [103] proposed an authentication method to decrease the number of unnecessary authentication during user handover in 5G networks. In this system, the Blockchain propagates the user's authenticity among other Software Defined Network (SDN) entities in the network. Moreover, Zhang et al. [104] proposed an authentication method for the seamless handover procedure in 5G networks. In this method, the user first registers in the network to insert a specific hash of the registration procedure into the Blockchain. Then, while the user is moving, this data would be used to manage the user's seamless connection.

DLT as a distributed database and verification solution: Xue et. al [105] proposed an authentication method to handle the user's movement in mobile vehicular networks. In this method, an intermediary smart contract is used to make a connection between foreign and home networks. The moving users receive a session key, which is also stored and managed by the smart contract, for further connections. Another method to provide mobility in the 5G networks is proposed by Lee et al. [106]. In this method, the authentication server sends the initial set of information to all base stations under its control. When a user joins one of the base stations, it sends the public key to the user and registers the connection in Blockchain. Then, the user sends her public key and timestamp to make connections, and then the base station sign and will broadcast these data.

Access point connection

DLT as a distributed database and verification solution: Sanda et al. [107] proposed a method in which the user installs "Auth-Wallet" to be verified by exchanging the "Auth-Coins" instead of her information. The user connects to the access point using its unique ID. The access point sends Auth-Coin to the user for verification and signing with the Bitcoin address. If the verification is successful, the token will be broadcast to the Blockchain and the user can be connected to the internet. Another system is proposed by Niu et al. [108] for Wi-Fi hotspot access. In the first step, the user requests a signature on Bitcoin address from the service provider by sending the real identity. Then, the digital signature would be sent to the user. Because the user's credentials are saved in the Blockchain when the user requests to connect to the network, the service provider and the Wi-Fi hotspot get valid credentials and provide the connection.

5.3. Proposed methods for smart healthcare

Access to health records

DLT as a distributed Database: Mohsin et al. [109] proposed an authentication method using RFID and finger vein (FV). In the first step, a hybrid, random binary pattern of the user's FV and RFID is derived and stored on Blockchain. The encrypted pattern is stored in an image with a steganography algorithm. When a user sends an authentication request to the access point, the FV and RFID features are extracted by reversing the AES method.

5.4. Proposed methods for WSNs

Mobility management

DLT as a distributed Database: BCTrust [110] is proposed for mobility management and aims to provide connectivity for WSNs in different clusters with only one authentication process. To do this, the first authentication controller (CPAN) stores the user's identification data on the Blockchain. Upon changing their cluster, the users' new CPAN sends a request to the Blockchain, and if the user has been authenticated before, the process is succeeded.

Network security

DLT as a distributed database and verification solution: Moinet et al. [111] proposed BATM in which each Network Node (NN) has cryptography keys, besides one master key for generating other keys for secondary encryption and digital signature. To connect to the network for the first time, NN issues a specific credential containing public keys to all other NNs. If authenticated NN includes the credential payload in a valid block, the authentication process is successful.

5.5. Proposed methods for ICNs

DLT as a distributed Database: Conti et al. [98] proposed BlockAuth to enable ***mobility management*** in ICNs. This system consists of global and local clusters and their associated ledgers. After the registration of the user using an Authorization Server (AS), this data is stored in the global Blockchain. Next, the user sends the same data to the Base Station (BS) for validation. The BS verifies the user's identity from the AS. A single authentication server in this method can be a single point of failure.

5.6. Proposed methods for Cloud computing

DLT as a distributed Database: several authentication methods aiming ***access management*** are proposed in cloud environment. Deep et al. [75] proposed a method to authenticate insider and outsider users. It checks the user's credentials and valid Blockchain node parameters to verify the user's identity. Another method, called SAMS [112], uses a master node to manage the security of the whole system. Before connection, the client creates a block and sends the nodes' and block's information to the master node. The master node creates a block with the received information to check the identity.

5.7. Proposed methods for general use-cases

Sign-on solution

DLT as a distributed Database: Wazid et al. [113] proposed a mutual authentication method in a crowd-sourcing environment. In this system, the user's credentials are stored in a tamper-proof ledger, and they will be authenticated by a central authority. AuthCoin [114] has four operational steps for authentication. First, each user generates a new key pair into Blockchain. Second, initial binding between the generated key pairs and their owner is established and stored in the Blockchain. Third, authentication is performed using challenge-response, and finally, all the logs are recorded into the Blockchain. User privacy and high resource consumption are two main challenges of this method.

DLT as a distributed database and verification solution: PTAS [115] is a mutual authentication prototype that works based on CertCoin[116] (i.e., a decentralized PKI on top Blockchain for website authentication). Firstly, Alice sends her identity and public key to Bob for authentication. Then, Bob sends a vector and a cube to m random full nodes. The full nodes retrieve all public keys that their vector and cube bits are *one*, then calculate a set of bits and send it to Bob. Bob performs exclusive-or of all received bits, and the final result is Alice's public key. Then, Bob sends an encrypted message to Alice to redo the same procedure.

Single sign-on solution

DLT as a distributed Database: Xiong et al. [84] proposed a privacy-aware authentication method supporting mutual authentication. To begin, a user registers in the system by sending her data to the nearest server. User data is added to both the smart cards and Blockchain. The verification process is done via consensus among Blockchain nodes.

5.8. Analysis and Discussions

As mentioned before, generally, using DLT for authentication can increase the integrity of the data, accountability of the users, and the difficulty of data falsification regarding credentials. Moreover, fully distributed implementation of an authentication method can improve the availability and fault tolerance of the system. In some specific use cases, DLT-based authentication can bring more unprecedented opportunities. For instance, healthcare data sharing using DLT would increase the user's sovereignty over their data and improve user privacy. Furthermore, providing mobility management via DLT-based authentication would eliminate unnecessary re-authentication procedures during user mobility, which can increase the system's performance.

Having these advantages in mind, the proposed methods mostly suffer from high computational time, transaction fees, and resource usage. Some studies use DLT as a database to store user credentials, which leads to inheriting the main problems of conventional centralized solutions. For instance, they have a central authority that will be a single point of failure, or it will decrease a system's availability in case of congestion. In addition, in several cases, the author's assumption about using the trusted server for authentication could be an obstacle

to the real-world implementation of the method. Moreover, using inappropriate consensus mechanisms in several systems can waste computing resources. Finally, user privacy remains an unsolved challenge for the majority of proposed methods.

Table 4 lists the features of existing methods, as well as their pros and cons considering privacy, availability, time consumption, cost-effectiveness, resource usage, resistance against attacks, and other issues specific to the methods in their proposed context. It is worth mentioning that the advantages and disadvantages of the methods have been gathered based on the author's comments, the other works which analyzed that specific system, and some other general concepts.

6. DLT-based Access Control Methods

This section presents the state of the arts in DLT-based access control methods. The categorization of the methods is the same as Section 5. Moreover, in the final part of this subsection, a general discussion about the DLT-based access control methods is provided.

6.1. Proposed methods for IoT/ smart cities

Right delegation

DLT as a distributed database: Ali et al. [117]proposed an access control solution focusing on the right delegation. Firstly, the owner deploys a smart contract containing the delegation policies. To validate the delegation requests, the owner sends that to the smart contract. Another work, named BlendCAC [118], implemented a CapBAC method in which the smart contracts store the access control matrix. The main challenge of this method is that a subject cannot obtain rights from more than one subject. This challenge is addressed in [119].

DLT as a distributed database and verification method: Tapas et al. [120] proposed an RBAC model for the right delegation in smart cities. At first, the user sends an access request to the Stack4Things (S4T) [121] delegation agent. If the validation is successful, S4T calls the Delegation contract to send a request to the Role contract. The access decision is based on the gathered data. Le et al. [122] proposed CapChain that allows users to delegate their access rights to IoT devices. To delegate the capability, the user A publishes a capability token using tx_1 and delegates the access to B (using tx_2). B sends a request to the device. The IoT device validates the request using CapChain.

Data sharing

DLT as a distributed Database: Kang et al. [123] proposed a data-sharing mechanism in vehicular edge computing and networks (VECON). In this system, vehicles generate and upload raw data in the Blockchain using a smart contract. For data sharing, the data requester lookup desired data through another smart contract. Then communicate with the data providers to apply for access authorization. Data audit and sharing of the records are done by the PoW model.

DLT as a distributed database and verification method: Sultana et al. [124, 125] proposed a data sharing and access control system via three smart contracts: ACC, RC, and JC. At first, the user sends an access request to the server to pass it to ACC. AAC registers the user in RC. Then RC verifies the user and their misbehavior from JC. If the user has a history of misbehavior, the request is rejected; otherwise, JC checks the user permission level and sends the result to ACC. A similar method is proposed by Zhang et al. [126]. Shafagh et al. [127] proposed a system with two data and control sub-layers. The transactions consist of data ownership streams, and corresponding access permissions. To share the data, the owner publishes a transaction in the Blockchain with a stream identifier and a public key. Then, the storage node checks the Blockchain for access rights to decide on the request.

Access to IoT sensors

DLT as a distributed Database: Dramé-Maigné et al. [128] designed an ABAC solution. In this method, administrators establish trust relationships for their devices, and the users deploy the attribute contract. When a user sends the access request, the target device connects to its gateway to retrieve attributes and evaluates the request against the policies. Another ABAC mechanism is proposed by Pinno et al. [129]. This system uses four separate Blockchains to store public credentials and relationships of all entities, contextual information, a history of connections to the object, and the authorization rules. When a user sends an access query, the decision engine gathers data from all Blockchains to validate the request.

DLT as a distributed database and verification method: Due to its flexibility, many researchers implemented *ABAC* solutions. In the proposed solution by Putra et al., [130], the smart contracts are responsible for authorizing the nodes based on their reputation. Zhang et al. [131] proposed an ABAC method in which the user, object, and authority node are three main actors. After receiving the user's access request, the object records the access information into the related smart contract and transmits the response to the user. The user signs the required information and sends it to the authority node. Finally, the authority node gets access credentials from the smart contract to validate the user identity. Fabric-iot [132] uses three kinds of smart contracts to store the URL of resource data, manage and store ABAC policies, and implement an access control method for non-admin users. In the proposed method by Ding et al., [133], the owner of the IoT device sends access policies to the Blockchain. The user chooses a satisfied subset of the policies regarding her needs. Then the owner checks the requestor's identity in the Blockchain and allows/denies the access request. Islam et al. [134] proposed another ABAC method in which creating an access policy and making the access control decision happens based on a consensus among all the stakeholders. Yutaka et al. [135] proposed a method that uses four smart contracts (i.e., PMC, SAMC, OAMC, and ACC). By receiving the access request, the ACC retrieves the subject and object attributes as well as the policy from the SAMC, OAMC, and PMC to perform access decisions. Moreover, [136] is proposed for RFID systems. Firstly, the RFID tag sends an access request to the RFID controller to redirect this message to the DApp.

Table 4: Comparison of existing authentication methods based on DLT

Auth. Type	*Approach*	*Refs.*	*App. Env.*	*Auth. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
Possession-based	Distributed database	[102]	*Communication networks*- ID generation	• Recording and generating user identification	Not mentioned	Permissioned	PBFT	• Proposing novel idea on IDaaS • Does not use pre-registered user information	• Does not protect user privacy • No security analysis
		[104]	*Communication networks*- handover	• Recording user's identification • Recording authentication logs	Not implemented			• Provides seamless handover • Low storage complexity • Perfect forward secrecy • Prevents Replay, password cracking, DoS/DDoS, and Sybil attacks	• None identified to date
		[103]	*Communication networks*- handover	• Recording user's identification	Ethereum	Permissioned	DPoS	• Provides acceptable handover delay • Protects user privacy • Resistant against DoS/DDoS, Spoofing, Link-ability, and Numb attacks	• None identified to date
		[114]	*All use-cases*- Access to the resource	• Recording user's identification • Recording authentication logs	Bitcoin/ Ethereum	Permissionless	PoW	• Protects integrity and availability • Tamper-proof challenge-response • Resistant against key injection • Resistant against password cracking, DoS/DDoS, and Sybil attacks	• High resource consumption • Does not protect the privacy (stores all authentication logs and results in the Blockchain) • Not sufficient security analysis
		[113]	*All use-cases*- Sign-in	• Recording user's identification	Not implemented			• Protects integrity and availability • Resistant against user impersonation, MitM, and Replay attacks	• Having a single point of failure
		[110]	*IoT (WSN)*- Mobility management	• Recording user's identification	Ethereum	Permissioned	Not mentioned	• Cryptanalysis is not possible • Energy efficient • Resistant against password cracking, DoS/DDoS, and Replay attacks	• System performance is not evaluated • Low scalability based on the context • Not sufficient security analysis
		[98]	*ICN*- Mobility management	• Recording user's identification	Not mentioned	Permissioned	Time based model [137]	• Resistant against password cracking, DoS/DDoS, Replay, Sybil, prefix hijacking, depending, and packet discarding attacks • The administrator cannot falsify the node's reputation	• Single authorization server can be a single point of failure • Not sufficient security analysis
		[75]	*Cloud*- Access to resources	• Recording user's identification	Not implemented			• Mechanism is robust and secure • Resistant against major DLT attacks • Resistant against password cracking, DoS/DDoS, MitM, Replay, and Spoofing attacks	• System performance is not evaluated
	Distributed database and verification	[107]	*Communication*- Connection to the internet	• Recording user's identification • User verification • Token issuing	Bitcoin	Permissionless	PoW	• Provides mutual authentication • Does not need user information • Protects user privacy [138]	• Uses Bitcoin address as the encryption key • No security analysis
		[105]	*Communication*- Mobility management	• Recording roaming session key • Token issuing	Ethereum	Permissioned	PoW	• Provides mutual authentication • Resistant against MitM, Replay, and modification attacks	• None identified to date

Auth. Type	*Approach*	*Refs.*	*App. Env.*	*Auth. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
Possession-based	Distributed database and verification	[87]	*IoT*- Access to IoT sensor	• Recording user's identification • User verification • Token issuing	Ethereum	Permission-less	PoW [139]	• Modification of the signed authentication message is impossible • High availability and scalability • Resistant against password cracking, MitM, and Replay attacks	• High computational cost • Not sufficient security analysis
		[106]	*Communication networks*-user mobility management	• Key handover by mining	Bitcoin	Permissioned	PoW	• Resilient against password cracking, MitM, Spoofing, re-synchronization, and rogue base station attacks	• None identified to date
		[111]	*IoT (WSN)*- Trust in WSN lifetime	• Recording user's identification • User verification	Bitcoin	Permission-less	PoW	• High availability • High scalability • Resistant against password cracking, DoS/DDoS, and MitM attacks	• High computational cost • Proposing no solution for storing private keys, make the method vulnerable to spoofing attacks
		[115]	*All use-cases*- Access to resources	• Server and user verification	Not implemented			• Provides mutual authentication • Resistant against 51% attack • Low computational overhead • Protects user privacy	• No security analysis
Knowledge-based	Distributed database	[108]	*Communication*- Wi-Fi hotspot access	• Recording user's identification • User verification	Bitcoin	Permission-less	PoW	• Provides accountability & anonymity • Provides suggestions for mutual authentication (without detail) • Uses quantum-safe Blockchain	• Not sufficient security analysis
Inherent-based	Distributed database	[99]	*IoT*-Access to IoT sensors	• Recording user's identification	Bitcoin	Permission-less	PoW	• Guarantees tamper-free credentials • Prevents data leakage • Uses biometric authentication	• High resource consumption • Not sufficient security analysis
multi-factor	Distributed database and verification	[101]	*IoT*- Provide trust in IoT in separate zones	• Recording user's identification • Ticket issuing	Ethereum	Permission-less	Not mentioned	• Protects data integrity & availability • Provides mutual authentication • High scalability • Resistant against DoS/DDoS, Replay, Spoofing, and Sybil attacks	• Not time efficient • Not cost efficient
	Distributed database	[100]	*IoT*- Access to IoT sensors	• Recording identification data • Recording authentication log	Eris [140]	Permissioned	PoS (Eris)	• Low resource consumption • Multi-factor authentication • Uses out-of-band authentication • Resistant against password cracking	• Single point of failure • Not sufficient security analysis
		[109]	*Smart healthcare*- Access to medical record via IoT	• User verification	Not mentioned	Permissioned	Not mentioned	• Protects user privacy • Resistant against password cracking and Spoofing attacks	• Not sufficient analysis on Blockchain size and scalability • Not sufficient security analysis
		[84]	*All use-cases*- single authentication	• Recording user's identification	Bitcoin	Permissioned	Ouroboros [57]	• Mutual authentication and privacy • Supports forward secrecy [141] • Resistant against password cracking, MitM, Replay, and Spoofing attacks	• None identified to date
		[112]	*Cloud*- Access to the resources	• Recording user's identification	Not implemented			• Impossibility of data falsification • Uses Multi-factor authentication • Resistant against MitM attack	• Single point of failure (master node) • Not sufficient security analysis

Next, DApp asks smart contracts about attributes and makes a decision based on them. Several papers implemented other access control methods. Banerjee et al. [142] proposed a multi-authority CP-ABE-based access control solution.

Sensor's data is encrypted using an access control policy. To access the data, the connected gateway to the IoT device creates a partial block with specific headers and sends the transaction to the Blockchain. The user who has the same access control policy can have access to the data through Blockchain. Albreiki et al. [143] proposed a CapBAC system in which oracles are gateways for connecting the IoT to Blockchain. When a user sends an access request to the IoT Data Access contract, if the request verification was successful, this contract forwards the request to the Aggregator contract to send it to a pool of oracles. Oracles retrieve requested data from off-chain storage and send the hash of data to the Aggregator to generate an access token and send it to the oracle and the user. Another system is proposed by Putra et al. [144]. In the access control process, the client or manager checks whether a particular component has permission to access a particular IoT device. Adding/removing devices in the manager network and defining access rules are done as smart contracts. In another method, IoTChain [145], firstly, the owner creates a smart contract for her data with an access policy and sends it to the Blockchain. When a user asks the authorization Blockchain to generate an access token, only if the validation was successful, the access token is generated. Novo [146] introduced an access control method focusing on scalability and energy consumption. To access the resource, the node sends an access request to Blockchain through the closest management hub. Once the miner informs the management hub about the access policy, it translates the answer back to the owner.

Network security

DLT as a distributed database and verification method: Tang et al. [147] proposed a cross-domain ABAC method. Trust Rule and Collaborative Rule contracts store permissions of the same domains and cross-domains, respectively. After receiving the access request, the object executes the smart contract to validate the request. Outchakoucht et al. [148] proposed a platform focusing on dynamic and distributed security policy based on machine learning techniques. When a user asks the data owner for access, the owner redirects this request to a smart contract that is trained to make particular decisions and learn from experience. A. Abdi et al. [149] proposed a light-weight hierarchical access control system using clustering concept. In this system an Edge Blockchain Manager is responsible for the authorization of local systems, Aggregated Edge Blockchain Manager controls different clusters and manages ABAC policies, and Cloud Consortium Blockchain Manager ensures that only authorized users access the resources. Hao et al. [150] proposed an intelligence access control architecture through a token accumulation mechanism.

6.2. Proposed methods for telecommunication

Resource management

DLT as a distributed database and verification method: Ghaffari et al. [151, 152] proposed an ABAC solution for internet-provisioning aims to eliminate the centralized access control of network and service providers. This method provides a secure payment for the network provider from the service provider account. So, they propose a novel business model for network/service providers. Ling et al. [153, 154, 155], proposed Blockchain Radio Access Network (BRAN) as a solution to implement self-organized access for users and providers, along with enabling mobility management. In this method, the user equipment and host access points agree on price and digitized spectrum assets via a smart contract. In their recent work, Ling et al. [156] proposed a Blockchain-based medium access control method.

Data sharing

DLT as a distributed database and verification method: Fan et al. [157] proposed a data-sharing scheme for Cognitive Cellular Networks (CCN) in 5G. Firstly, the content provider stores the data and access permissions in the Blockchain. The encrypted hash of data is stored in the cloud. When the requester wants to access the data, miners assess the request, and after consensus, the user connects to the content provider to get data.

6.3. Proposed methods for Cloud computing

Data/resource sharing

DLT as a distributed Database: Qin et al. [158] proposed an ABAC method to share data in the cloud environment. In this system, the CA is responsible for managing the security of the whole system. Firstly, the CA issues an attribute key to the user and CSP in the smart contract, having an expiration time. Then, in the access control phase, the data owner first uploads the ciphered text to the CSP, and the CSP invokes the contract to obtain the user's valid attribute set. If the user is valid, she can perform the final decryption to get the desired information. Alansari et al. [159], proposed an ABAC method for cloud federation. In this system, federated cloud organizations can define attribute-based rules and store them in the Blockchain to provide fine-grained secure data sharing for the users.

DLT as a distributed database and verification method: Guo et al. [160] proposed a traceable attribute-based encryption method (TABE-DAC) that uses the ABE solution to provide the capability of sharing private data in the cloud. Yang et al. [161] proposed AuthPrivacyChain in which policies and access logs are stored in Blockchain, and access management is done by the smart contract. The user sends an access request to the cloud to decrypt it and sends another request based on the first one to the Blockchain. The smart contract answers the request based on access permissions. PrivacyGuard [162] is another system that focuses on user and data privacy on the cloud. In this system, data is generated by the owner and encrypted and stored in cloud storage by a trusted agent. Owners can define their access policies in smart contracts. The user invokes the owner's contract to ask for permission, data access rules, and deposit payment. TBAC [163] is an ABAC solution for resource sharing that exploits four types of transactions to record the information of subjects and objects, send the access request, and access decision.

Data/resource access

DLT as a distributed database and verification method: Sukhodolskiy et al. [164] presented a system that manages the user access via smart contracts, containing the location of the object, access policy, and additional owner's information. One obstacle in the adoption of this method is the incompatibility between the immutability of typical Blockchains and the attribute updates/revocations of ABE, which is addressed in [165]. Wang et al. [166] proposed a fine-grained access control for cloud storage. In this method, the owner deploys a smart contract to store the essential data of the file. To grant access, the owner defines the expiration time and a secret key and adds them to the smart contract. Then sends these data to the user. Finally, the user can download and decrypt the file.

6.4. Proposed methods for smart healthcare

Data sharing

DLT as a distributed database: Zhang et al. [167] proposed a hierarchical model for sharing the healthcare data. In this method, Blockchain acts as a the distributed ledger of permissioned clients to store verified codes of ciphertexts (keys) and record the hash values of auditing logs.

DLT as a distributed database and verification method: Rajput et al. [168] proposed Blockchain-based smart healthcare data sharing system. In this method, after registration, the emergency doctors can retrieve the patient's data by sending access requests and the patient's ID via Blockchain and smart contract. The smart contract sets an expiration time for the request. Nguyen et al. [169] use Blockchain for sharing patient's healthcare data on the Internet of Medical Things (IoMT) networks. In this method, the requested data is stored in IFPS, and using smart contracts, the system shares the required data with the eligible user.

Data access

DLT as a distributed database and verification method: Li et al. [170] proposed a system based on certificate-less cryptography. In this system, the data owner creates an ACL and then stores it in the Blockchain. When a user wants to access the data, sends a request transaction to be verified by miners regarding the transaction's validity and user's ID. If both validations were successful, access is granted.

6.5. Proposed methods for ICNs

DLT as a distributed database and verification method:

SBAC [171] aims to achieve hierarchical access by proposing an ABAC method for ***data sharing*** in two levels: 1) matching-based access control model and 2) Blockchain-based access token mechanism. In the first level, the content provider defines a set of attributes, and in the second level, it generates an access token for the requester in Blockchain.

6.6. Proposed methods for general use-cases

In this subsection, we introduce several DLT-based access control methods that are proposed as a general solution. These methods can be categorized into right delegation, access management, and data sharing.

Right delegation

DLT as a distributed Database: Masea et al. [172] proposed an ABAC method to transfer access rights among users on the Bitcoin network. The owner defines the delegation policies and stores its external storage link in the Blockchain.

Data/resource access management

DLT as a distributed Database: Shafeeq et al. [173] proposed an ABAC mechanism using Tangle [174] DAG. In this method, the owners define and manage the access rules, security policies, and authorization granularity over their assets and store them in DLT. Moreover, they send an authorization token to the requesters. A similar solution is proposed in BRIGHT [175]. Furthermore, Ihle et al. [176] proposed an RBAC model that saves all the subject roles in the key-value data model on the smart contracts. In addition to general solutions, in multi-administrative environments, several security problems such as transitive access and access in conflict of interest domains are addressed in the latest research. Ali et al. [177] proposed BCON, to address the security problems of transitive access among multiple conflicts of interest domains. In this method, users' access histories and transitive access are stored on Blockchain. The authorization decision is based on Blockchain endorsement that transitive access will not occur. Using group-based policy, Paillisse et al. [178] proposed an access control solution consisting of three layers to set policies, store the access data and ensure its integrity in Blockchain, and so provide a connection between resource and Blockchain.

DLT as a distributed database and verification method: Rouhani et al. [179] proposed an ABAC system consisting of Policy Information Point (PIP), Policy Decision Point (PDP), and Policy Administration Point (PAP) implemented as smart contracts. In this system, the evaluation engine takes policies from Blockchain to make access decisions. Similar solutions are proposed in [180, 181]. Wang et al. [182], proposed Attribute-based Distributed Access Control (ADAC). After receiving the request, Access Control Contract obtains attributes from the Subject Contract, Object Contract, and Blockchain. Then, it obtains the policies from the Policy Contracts and finally makes an access decision. SC-RBAC [183] is an RBAC method that consists of three different smart contracts responsible for handling the user and role permissions, creating/ changing/ disabling the roles, and managing the user's access by creating, enrolling, disabling the user, or changing his role. Another RBAC method is RBAC-SC [184] which consists of a smart contract and a challenge-response protocol. The smart contract creates, changes, and revokes the user's role assignments. Zyskind et al. [185, 186] proposed Enigma for ACL-based access management and log audition. To access an object, the subject sends his signed request to the ledger to be compared with the existing ACL. Only if the user has sufficient access rights, the connection is established. In their other work, Zyskind et al. [187] proposed a permission management system focusing on user privacy. Kiran et al. [188] use two smart contracts to handle access policy definition and access management. Policy Definition contracts have an 'owner' to manage access permissions, and also store a pointer to the restricted off-chain data. When the requester sends the request to the Data Access contract, it only returns the data that the requester is allowed to use.

Data sharing

DLT as a distributed database and verification method: Bowen et al. [189] proposed a data distribution system in which Blockchain maintains the complete transaction records. The Data Creator (DC) generates a public key, master private key, and access control strategy, and sends it to the Blockchain. Then, it deploys a smart contract with the user to grant attributes. The user sends her public key to the smart contract and gets the attribute's private key. Using this contract, DC generates the required key and sends it to the user. Another similar system is proposed by Wu et al. [190]. This system is a traceable attribute-based encryption method that sends encrypted policies by attribute filter to achieve fine-grained access control on user data. Gao et al. [191] proposed TrustAccess, to improve the work proposed in [190]. Wang et al. [192] proposed a fine-grained access control method. First, the owner encrypts the system's master key and saves it to the Blockchain, then deploys a smart contract. The user sends the registration request to the owner; the owner retrieves the secret key for the user and saves it on the Blockchain, and sends the transaction ID and smart contract's address to the user. These data will be used for the next connections.

6.7. Analysis and Discussion

A comparison of the existing access control methods based on different features is provided in Table 5. This table also lists their advantages and disadvantages regarding privacy, availability, time consumption, etc. It is worth mentioning that the advantages and disadvantages of the proposed methods have been listed using the same method as in Section 5.8.

The most significant benefit of using DLT in the access control process is to increase the immutability, integrity, and availability of the rules, data, and services. Two of the most highlighted features of smart contracts that encourage the authors to use this technology for AAC procedures are the public availability of the code and data and the fact that the code is always the right paradigm. When a method uses DLT for access management, the system availability is guaranteed by removing the single point of failure. In addition, the service and maintenance costs can be reduced by removing the need for a third party.

On the negative side, DLT can be problematic for resource-constrained devices, such as those in the IoT. In some cases, the adjusted version of a consensus model has been used to decrease resource consumption, but this modification brings with it several concerns regarding security and immutability. Another challenge for these methods is the high time consumption compared to centralized systems. Meanwhile, proposing auditable access control methods can violate user privacy. Similar to authentication methods, using Blockchain exclusively as a distributed database can result in no resolution of the main issues of the conventional methods, such as a single point of failure. The other two significant challenges in these systems are the size of the blocks and required storage, as the system performance can be negatively influenced by an oversized chain [138]. A comprehensive discussion on the limitation of DLT for AAC is provided in section 8.

7. DLT-based comprehensive AAC methods

As mentioned earlier, AAC is a complete procedure to manage secure user access to resources. While most of the works in this field focus on only one aspect, some do indeed propose and evaluate a comprehensive AAC procedure. This section introduces the current AAC methods based on DLT.

7.1. Proposed methods for IoT/ smart cities (Data access)

DLT as a distributed Database: Almadhoun et al. [193] defines on-chain and off-chain activities in the authentication procedure. In the on-chain part, an administrator creates a smart contract, defines the user permissions, registers the IoT device, assigns an Ethereum address, and maps it to a fog node. In the access control step, the user sends a request to the smart contract to check the permissions and provide an access token.

DLT as a distributed database and verification method: Widick et al. [194] proposed a method containing two smart contracts to handle digital certificates, and access control procedures. The authentication step is implemented via a smart contract called TTP. Firstly, the user sends her request to TTP and TTP sends the certificate after validating the user's data stored in DLT. After that, an access token will be issued and handled by AC smart contract. This token includes the details of the user, the access duration, and the resources of the task. FairAccess [195, 196] is one of the first proposed AAC solutions. In the authentication step, the system uses an access token created based on the user's credentials. The access control step implements RBAC solutions, in two central and distributed levels. For access granting, the sender and receiver of a token must solve a cryptographic problem. Gauhar et al. proposed xDBAuth [3] by introducing two access domains internal and external. In the authentication step, they propose a new consensus model to find the platform hash for a given pseudonymous ID stored on the local Blockchain. Then, a central Blockchain manager and two smart contracts handle the access control.

7.2. Proposed methods for smart healthcare (Data access)

DLT as a distributed database and verification method: Akkaoui et al. [197], proposed EdgeMediChain. For authentication, the registered user's data is verified by the edge nodes in Blockchain, and only after that, the user can start sending related data to be further processed. Access control in EdgeMediChain is based on the RBAC model and is handled by the RBAC contract. If the requestor's permissions to access the data are not null, the access permission will be granted.

7.3. Proposed methods for general use-cases

Data access

DLT as a distributed Database: BSeIn [73] is a mutual authentication and fine-grained authorization mechanism. Mutual authentication in this system is provided by a one-time public/private key pair for each request. After successful authentication to the system, when the user aims to access an object's data, invokes a smart contract, called PDHT, to get the object's desired rules. If the user's attributes satisfy the rules, then she

can publish the reasonable request. To broadcast the request to the Blockchain, the user generates a public and private key and prepares the request based on her needs.

Single sign-on

DLT as a distributed database and verification method: Zhang et al. [74] proposed a general-purpose framework to manage the different permissions based on user's related data for different websites. Firstly, the user stores her identity in the Blockchain and her encrypted personal data in the off-chain storage. To prepare different websites with the user's data, a smart contract is attached to the user's identity. When a user sends a login request to a website, the service provider verifies the identity of the user and retrieves the user's data from the off-chain storage based on the rules in the smart contract.

7.4. Analysis and Discussion

DLT has been used for a complete authentication and access control procedure in several works. We compared the existing AAC methods based on features listed in Table 6. This table presents the pros and cons of existing methods, considering privacy issues, availability, time consumption, cost-effectiveness, resource usage, and other upcoming challenges specific to each proposed method in their targeted context. It also summarizes the security features or concerns of the existing methods in terms of security and attacks.

In a high-level abstract, using DLT in authentication and access control methods have the same advantages mentioned in Sections 5.8, 6.7. However, most of these systems require high computational resources, time, and transaction fees. Several solutions used central databases to decrease the required storage, a solution that leads to have a single point of failure in the system. User privacy remains an unsolved challenge for several methods.

8. Lessons learned and discussion

The concerns about existing methods and our lessons learned are provided in the following sections. The concerns highlighted here are interpreted based on the analysis of the surveyed systems and the nature of DLT.

8.1. Privacy

Several use cases such as telecommunications, cloud computing, IoT, and smart healthcare are highly sensitive domains for protecting users' privacy because they involve the users' personally identifiable information (PII). Privacy provisioning increases the user's ownership of their data and eliminates uninvited surveillance. Data leakage due to a bad decision in an AI system of cloud computing [25], curious trusted third parties in the IoT [198], non-secure communication interfaces, and irresponsible or unethical administrative actions in telecommunication or healthcare [25] are some of the data breach scenarios.

One of the intrinsic characteristics of DLT is its meta-data privacy-preserving [25]. It means the real-world identities of the sender and receiver of a transaction are both masked (i.e., the privacy of the transaction's metadata is provided). However, from another perspective, the AAC procedure may require the user's identity (i.e., their private information) to be sent to the DLT network for decision-making. Indeed, the privacy of the transactions' contents is not offered in this technology. So, providing the AAC with zero knowledge (or minimum required knowledge without revealing the PII) is required in DLT-based AAC systems.

Privacy faces other challenges when analyzing different DLT types. In permissioned DLTs, which are designed to serve the needs of one/several enterprises, the user's privacy can be violated with minimum effort. In these systems, entities need to trust each other (because the communication network is small-scale), and so providing trust can violate the privacy and anonymity of the individuals. On the other hand, using permissionless DLT models increases the latency and storage complexity.

A decentralized identity management system based on DLT (i.e., Self-Sovereign Identity(SSI)) [81] is gaining more attention as an option to resolve the above-mentioned issues. Self-sovereign identity is a novel paradigm of decentralized identity management, in which all entities can manage their PII by storing it in their preferred devices and selectively granting access to trusted third parties, without referring to any intermediary to validate these claims. SSI systems can manage the user's private data and reveal the required non-PII information to provide authentication (and authorization). These systems can therefore offer transparency (i.e., all nodes would participate in the AAC procedure), privacy (i.e., no one can find out the users' real-world identity), and user-centric data ownership (i.e., users have more control over their data and can reveal it according to their preference).

8.2. Latency and Storage

DLT is an append-only technology in which transactions can be added to the system after consensus among all or majority of participating nodes in the network. Generally, the requests in DLT-based networks can be categorized into two main groups:

1. *Calls* in which the sender only aims to retrieve the value(s) from the ledger. These kinds of requests do not change the state and would not be added to the distributed ledger;
2. *Transactions* update the network's state. So, they need to be added to the transaction's pool, validated, and fitted into the blocks after consensus.

Calls do not increase the storage complexity of a system. Moreover, from the latency viewpoint, since data is stored in distributed nodes, the latency of reading operation decreases significantly (compared to centralized SQL) [25]. Thus, DLT can be more beneficial for the "read" operation in AAC.

For the transactions, extra storage would be required in all the nodes that have a copy of the ledger. Transactions increase the storage complexity of the whole system. So, the burden of DLT ledger storage, which is distributed among nodes, can cause a resource problem for the users who contribute to the system [199, 200]. As for the latency, transactions take at least one "block time" to be confirmed and added to the block. Therefore, because of the consensus among nodes in

Table 5: Comparison of existing access control methods based on DLT

A.C. Type	*Appro.*	*Refs.*	*App. Env.*	*A.C. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
Generic (support all mechanisms)	DDB*	[175]	*All use-cases*-resource access	• Storing rights and rules	Bitcoin	Permissioned	Modified PoW	• Right management for owners • Flexible security and cost level	• No incentive for mining [168] • High delay in access
		[117]	*IoT/ Smart City*-right delegation	• Policy Storing • Stores access logs	Hyperledger Fabric	Permissioned	PBFT	• Low resource consumption • High efficiency	• Saves huge amounts of data in Blockchain (low scalability)
		[167]	*Healthcare*-Data sharing	• Policy storing • Recording logs	Ethereum	Permissioned	PoS	• Low time consumption	• None identified to date
		[177]	*All use-cases*-access delegation	• Access log recording	Hyperledger Fabric	Permissioned	PBFT	• Protects against transitive access • DDoS resistance	• None identified to date
		[178]	*All use-cases* Single access	• Policy storing	Hyperledger Fabric	Permissioned	PBFT	• High scalability • Low storage consumption	• Performs the queries based only on exact parameters matches
		[123]	*IoT*- Data sharing	• Recording data and vehicle information	Not mentioned	Permissioned	PoStorage & PoW	• Prevent second-hand data sharing without authorization • Protects privacy	• High resource consumption
	AC Mng.**	[156]	*Communication*-Resource access	• Medium access validation	Ethereum	Permissioned	PoD	• Prevents rogue devices from exhibiting selfish behaviors	• The method inherits the problems of [153, 154, 155]
	DDB & AC Mng.***	[153, 155, 154]	*Communication*-Resource access	• Policy storing • Access validation • Definition of access policies • Access decision enforcement	Ethereum	Permissioned	PoD [139]	• High efficiency	• Low scalability • Low service quality
		[150]	*IoT*- Resource access	• Policy storing • Access validation • Definition of access policies • Access decision enforcement	NaN	Consortium	PBFT	• Protects user privacy • Provides trust	• None identified to the time
		[187]	*General purpose*- Resource access	• Policy storing • Access validation	Self-Deployed [12]	Permissioned	PoW	• Minimizes system load • Protects user privacy • Resistant to Sybil attacks [12]	• Low Scalability [201] • Having a single point of failure • High energy consumption
		[127]	*IoT*- Data sharing	• Access right storing • Access validation	Bitcoin	Permissioned	PoW	• Preserves user privacy	• No performance analysis • PoW for IoT is not effective
		[145]	*IoT*- Access to IoT sensor	• Policy storing • Token issuing • Access policy definition	Ethereum	Permissioned	Proof-of-Possession (PoP)	• Uses encrypted storage • Protects user privacy • Resilient to DoS and MitM attacks	• Depends on an intermediary entity for key distribution [191]
		[161]	*Cloud*- Access to cloud resource	• Permission storing • Access validation, • Access log recording	EOS (Kylin/ Jungle)	Permissioned	Not mentioned	• Protects user privacy • Resistant against internal attacks • Confidentiality, integrity, availability, authenticity, and accountability are provided	• Does not provide any security solution for secret key sharing

A.C. Type	*Appro.*	*Refs.*	*App. Env.*	*A.C. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
		[144]	*IoT*- Access to IoT sensor	• Storing policies • Access validation • Access decision enforcement • Access policy definition	Ethereum	Permissioned	Different methods	• High scalability for IoT devices	• High latency
		[188]	*All use-cases*- Data access	• Policy definition • Access validation • Access decision enforcement • Access policy definition	Ethereum	Permissioned	Not mentioned	• Protects user privacy • Uses distributed off-chain storage to decrease system load	• Stores plain text data in storage
		[125, 124]	*IoT*- Data sharing	• Policy storing • Access validation • Access log recording	Ethereum	Permissioned	Not mentioned	• High reliability • Trustworthy system	• Does not protect user privacy
		[169]	*Smart healthcare*- Data sharing	• Policy storing • Access validation • Access log recording • Access policy definition	Hyperledger Fabric	Permissioned	PBFT	• Data privacy	• None identified to date
		[168]	*Smart healthcare*- Data sharing	• Policy storing • Access validation • Access log recording • Access policy definition	Hyperledger Fabric	Permissioned	Not mentioned	• Data privacy	• Insufficient security analysis
ABAC	DDB	[172]	*All use-cases*- Access right transferring	• Policy storing	Bitcoin	Permissionless	PoW	• Stores compressed data in DLT • Resistant against to DDoS and MitM attack [171]	• High transaction fee and delay [201] • Privacy is not protected • There are two single points of failures
		[181]	*All use-cases*- Data access	• Policy storing	Ethereum	Permissionless	Not mentioned	• Stores compressed data in Blockchain	• Saves new record for each rule • High transaction fees & delay [201] • Does not protect privacy
		[158]	*Cloud*- Data sharing	• Policy storing	Ethereum	Permissioned	Not mentioned	• Low computational cost • ABE provides confidentiality	• The CA is a single point of failure
		[173]	*All use-cases*- Access to resource/data	• Policy storing • Authorization granularity level storing	Tangle	Permissioned	FPC [202]	• Protects user & policy privacy • Provides MAM • High scalability	• None identified to date
		[159]	*Cloud*- data and resource sharing	• Storing of policies and attributes	Proposed as a framework			• Provides secure resource sharing • Protects data and user's privacy	• Low scalability
		[129]	*IoT*- Access to IoT sensor	• Policy and credential storing • Access logs recording	Not mentioned			• Compatible with a wide range of access control mechanisms • High scalability	• Four Blockchains must be used [131] • Efficiency is not proven [203] • Access engine is SPoF

A.C. Type	*Appro.*	*Refs.*	*App. Env.*	*A.C. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
ABAC	AC Mng.	[171]	*ICN*-Data sharing	• Access validation	Ethereum	Permissioned	PoS	• Multi-level content access • Resistant against Cache poisoning, DDoS, and MitM attacks	• Low scalability [179]
		[180]	*All use-cases*-Data access	• Access validation • Access decision enforcement	Ethereum Ropsten	Permissionless	Not mentioned	• Stage effective • Attribute management	• User privacy remained a challenge
		[179]	*All use-cases*-Data access	• Access validation • Access policy definition • Access decision enforcement	Hyperledger fabric	Permissioned	Raft and Kafka	• High scalability • Protects user and data privacy	• None identified to date
	DDB & AC Mng.	[128]	*IoT*- Access to IoT sensor	• Access validation • Attribute storing	Not mentioned			• Multiple domain access • Scalable and flexible • Mitigates reply attacks	• High time consumption • Information leakage is not addressed • Lack of implementation details
		[126]	*IoT*- Data sharing	• Policy storing • Access validation • Access log recording • Access policy definition • Access decision enforcement	Ethereum	Permissioned	PoW	• Implements trustworthy access control for IoT systems using smart contracts	• Does not protect user privacy • High transaction fees [124, 201] • Limited environment attributes
		[133]	*IoT*- Access to IoT sensor	• ID storing • Access verification	Hyperledger Fabric	Permissioned	PBFT	• Avoids data tampering • Lightweight calculation	• High message passing in the network • No efficient consensus [124]
		[149]	*IoT*- network security	• Policy storing • Access verification • decision enforcement • definition of access policies	HyperLedger fabric	Permissioned	PBFT	• Clustering to improve scalability	• None identified to date
		[151, 152]	*Communication*-resource provisioning	• Policy storing • Access verification • decision enforcement • definition of access policies	Ethereum	Permissioned	PoW	• Provides flexible access control • Implements secure payment • High scalability	• Storage effectiveness is not assessed
		[130]	*IoT*- data access	• Policy storing • Access verification • Access policy definition	Rinkbey	Permissionless	PoA	• Provides flexible access control • Resistant against self-promoting, and Ballot-stuffing attacks.	• Low scalability • Attribute authority is a single point of failure
		[132]	*IoT*- Access to IoT sensor's data	• Stores policy & URL • Access verification • Access policy definition • Access decision enforcement	Hyperledger Fabric	Permissioned	Kafka	• Lightweight computation • Dynamic permissions	• Performance is not proven • Low scalability

A.C. Type	*Appro.*	*Refs.*	*App. Env.*	*A.C. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
ABAC	DDB & AC Mng.	[163]	*Cloud*- Cloud resource sharing	• Rule storing • Access validation • Access log recording		Not mentioned		• Supports flexible and diverse permission management	• Performance is not proven
		[131]	*IoT*- Access to IoT sensor	• Policy storing • Access validation • Access policy definition • Access decision enforcement	Hyperledger Fabric	Permissioned	Not mentioned	• Resistant against collision and reply attacks • High scalability	• Does not protect user privacy
		[136]	*IoT*- Access control of RFID nodes	• Attribute storing • Access validation • Access policy definition	Ethereum	Permissioned	PoW	• Resistant against message substitution, and reply attacks • High scalability	• High resource consumption
		[162]	*Cloud*- Data sharing	• Policy & log storing • Access validation • Access policy definition • Access decision enforcement	Ethereum	Permissioned	PoA	• Protects user and data privacy • Provides secure payment	• Requires specific hardware [204]
		[182]	*All use-cases*- Resource access	• Rule storing • Access validation • Access policy definition • Access decision enforcement	Ethereum (Ropsten)	Permissioned	Not mentioned	• Low resource consumption	• Low maintainability [205, 206] • Does not protect user privacy
		[134]	*IoT*- Access to IoT device	• Access validation • Access log recording	Hyperledger fabric	Permissioned	PBFT	• High scalability • High performance	• High latency
		[135]	*IoT*- Access to IoT device	• Policy storing • Access validation • Access policy definition • Access decision enforcement	Ethereum	Permissioned	Not mentioned	• Acceptable scalability	• High latency for access control • Does not protect user privacy
		[157]	*5G*-Data sharing	• Policy storing • Access validation		Not mentioned		• Protects providers' privacy • Provides forward secrecy	• Low scalability (because of block size limitation)
		[147]	*IoT*-Providing trust	• Policy storing • Access validation • Access policy definition • Access decision enforcement		Not mentioned		• Secure interactions [207] • Uses an incentive mechanism	• No implementation • No analysis
		[148]	*IoT*-Network security using ML	• Access validation • Access log recording		Not mentioned		• Novel concept for machine learning and dynamic access control	• False decisions in first steps • No implementation

A.C. Type	*Appro.*	*Refs.*	*App. Env.*	*A.C. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
CapBAC	DDB	[118]	*IoT*- Access right delegation	• Policy storing • Storing capabilities • Access policy definition	Ethereum	Permissioned	PoW	• High scalability • Hierarchical delegation	• A subject can only obtain rights from one subject [119] • High resource consumption
		[119]	*IoT*- Access right delegation	• Records capability tokens & access matrix • Access policy definition	Ethereum	Permissioned	Not mentioned	• Supports multi-delegation [208] • No limitation in the right delegation	• Tokens are stored in Blockchain with no encryption
	DDB & AC Mng.	[122]	*IoT*- Capability delegation	• Stores capabilities • Access validation	Monero	Permissioned	PoW	• Protects privacy for capabilities and users via obfuscation	• High latency • High resource consumption
		[143]	*IoT*-Access to IoT device's	• Stores reputation scores • Access validation • Access policy definition • Access decision enforcement	Ethereum	Permissioned	Customized	• Open-source implementation • Resistant against MitM attacks	• None identified to date
RBAC	DDB	[176]	*All use-cases*- Access to data	• Storing policies and rules • Access policy definition	Hyperledger Fabric	Permissioned	Not Mentioned	• Can be used by all types of DApps	• User ID is a key and can be forged
	DDB & AC Mng.	[120]	*IoT*- Access delegation	• Access validation • Access policy definition	Ethereum	Permissioned	PoW	• High fault tolerance • Time efficiency	• High energy consumption
		[183]	*All use-cases*- Resource access	• Storing roles • Access validation	Ethereum	Permissioned	Not mentioned	• Compatibility with DApps • Flexible interfaces	• Low performance • Low reliability
		[184]	*All use-cases*- Resource access	• Policy storing • Access validation	Ethereum Ropsten	Permissionless	PoW	• Comprehensive AAC method • Open-source implementation	• High computation cost
ACL	DDB & AC Mng.	[170]	*Smart healthcare*- Access to EHR	• Policy storing • Access validation	Not implemented			• High traceability, and scalability • Certificate-less cryptography	• Lack of details on its implementation in Blockchain
		[185]	*All use-cases*- Access to data/resource	• Access validation • Records access log	Customized Blockchain with financial incentives			• Protects user privacy • Secure multi-party computation • High scalability	• Uses self-designed Blockchain without performance analysis
		[146]	*IoT*- Access to wireless sensors	• ACL storing • access policy definition	Ethereum	Permissioned	Not mentioned	• Low energy consumption • High scalability • Low latency	• Low performance because of RPC • Single management hub [124]
CP-ABE-based	DDB	[192]	*All use-cases*- Data Sharing	• Storing secret keys and IDs	Ethereum	Permissionless	PoW	• Protects user privacy • Shares encrypted data	• No secure algorithm for master key sharing
		[160]	*All use-cases*- Resource access	• Storing access credentials	Eclipse	Permissioned	Not mentioned	• Protects data privacy • Policy updates capability	• Has a single point of failure (Authority)

A.C. Type	*Appro.*	*Refs.*	*App. Env.*	*A.C. step*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
CP-ABE-based		[142]	*Industrial IoT*- Access to sensor data	• Storing encrypted data	Not mentioned	Permissioned	PBFT	• Protects data privacy • Protects policy privacy • MitM and Reply attacks resistant	• Attribute authority is a single point of failure
		[189]	*All use-cases*- Data sharing	• Access log recording		Not implemented		• Protects data and user privacy • Data tracking	• Has the single point of failure • No implementation
		[190]	*All use-cases*- Data sharing	• Access log recording		Not implemented		• Protects data and policy privacy • Data tracking • Security analysis	• Has the single point of failure • Low decryption performance [124] • Lack of implementation details
	AC Mng.	[191]	*All use-cases*- Data sharing	• Access validation • Access decision enforcement	Not mentioned	Permissioned	PBFT	• Protects attribute/policy privacy	• Low efficiency
	DDB & AC Mng.	[166]	*Cloud*- Access to cloud storage	• Storing ID files • Access validation • Access log recording	Ethereum	Permissioned	Not mentioned	• Uses cloud-based off-chain storage • Low latency	• None identified to date
		[164]	*Cloud*- Access to cloud resources	• Access validation • Access log recording • Access decision enforcement	Ethereum	Permissionless	Not mentioned	• Access policy customization	• Incompatibility with the immutability of typical Blockchains
		[165]	*All use-cases*- Data sharing	• Access validation • Access log recording	Ethereum	Permissioned	PBTF	• Solves incompatibility in [164] • Update-oriented access control • Security analysis	• None identified to date

*DLT is used as only Distributed DataBase.
**DLT is used as a solution for ACcess Management.
***DLT is used as both Distributed DataBase and a solution for ACcess Management.

Table 6: Comparison of existing comprehensive AAC methods based on DLT

Refs.	*Auth. Type*	*A.C. Type*	*Approach (A. C.)*	*Appro. (Auth.)*	*Auth./A.C. step*	*App. Env.*	*BC Plat.*	*BC Type*	*Cons. Model*	*Pros*	*Cons*
[195, 196]	Possession-based	ABAC	Distributed database & Access Management	Distributed database and verification method	★ Token issuing + ★ Recording logs ◇ Policy definition ◇ Policy storing ◇ Token validation + ◇ Recording logs	*IoT*- Access to IoT sensor	Bitcoin	Permissioned	PoW	• Preserves privacy • Resistant against DDoS and MitM attacks [171]	• Renewing the expired token is not considered • High delay [201] • High transaction fee [201]
[193]		General	Distributed database		★ Record Identification ★ User verification ◇ Policy definition ◇ Policy storing	*IoT*- Access to IoT sensor	Ethereum	Permissionless	Not mentioned	• Open-source • Low resource consumption • Password cracking, MitM, DoS, Replay, and Spoofing attacks resistant	• User privacy is not provided • Does not offer mutual authentication

[74]	General	Distributed database and access management	Distributed database and verification method	⋆ Recording identifications ⋆ User verification ⋄ Policy definition ⋄ Policy storing ⋄ Access verification ⋄ Access decision enforcement	*All use-cases*- single access to different websites	Ethereum	Permission-less	Not mentioned	• Distributed off-chain storage • Provides different permissions based on the website • Password cracking, and DoS attacks resistant	• Trusted server is the single point of failure
[3]				⋆ Recording identifications ⋆ User verification ⋄ Policy definition ⋄ Policy storing ⋄ Access verification ⋄ Recording logs	*IoT*- Connection to IoT network and access to IoT sensor	Customized Blockchain	Permissioned	PoAI	• Off-chain storage can decrease chain size • High fault tolerance • Reliable user privacy • Resistant against password cracking, DoS/DDoS, Spoofing, and Sybil attacks	• Off-chain storage can be a single point of failure • Does not offer mutual authentication
[194]				⋆ Token and certificate issuing ⋆ Recording logs ⋄ Policy definition and storing ⋄ Access verification	*IoT*- Access to sensors' data in smart cities	Ethereum Ropsten	Permission-less	Not mentioned	• Guarantees tamper-free credentials	• Single point of failure • No security analysis
[197]	RBAC			⋆ Recording identifications ⋆ User verification ⋄ Role definition and storing ⋄ Access verification	*smart healthcare*- Access to health records	Ethereum	Permissioned	PoW & PoA	• Resistant against 51%, double spending, Password cracking, DoS/DDoS, and Replay attacks • Provides role assignment capability • Protects user privacy by removing PII data	• Privacy preservation algorithm is vulnerable to linking attacks • Not sufficient security analysis
[73]	Fine-grained		Distributed database	⋆ Recording user's identifications ⋆ Recording authentication log ⋄ Role definition and storing ⋄ Access verification	*Industry 4.0*- Data access	JUICE	Permissioned	PBFT	• Mutual and anonymous authentication • Protects the user's privacy • Password cracking, MitM, DoS, Replay, and Spoofing attacks resistant	• Not identified to date

+ Note that, (⋆) bullet points define the authentication steps, and (⋄) bullet points are access control steps

DLT, the transaction's latency is generally higher than in conventional systems. Since the latency is dependent on the complexity of the consensus puzzle and on the size of the block (i.e., the maximum number of transactions that can be fitted into one block), this parameter is adjustable as a trade-off between security and latency.

To summarize, the existing methods that use DLT as the distributed database are generally more efficient regarding latency and storage. However, they mostly suffer from having a single point of failure and from inheriting the problems of existing centralized systems.

On the one hand, methods which use DLT as a database and for making access decisions are more reliable regarding the existing vulnerabilities of conventional systems. On the other hand, they have higher complexity regarding storage and latency. However, system designers can reach the required latency by combining centralized and distributed systems. Moreover, they can adjust the performance by selecting high-performance DLT platforms and tuning the latency based on the block size and time. In this regard, a key challenge would be to design a secure system against DLT's intrinsic vulnerabilities such as 51% attacks. Decreasing the complexity of the consensus puzzle increases the possibility of putting the system's security at higher risk.

8.3. Transaction cost

The user needs to pay a predefined fee to submit the transactions to a DLT network (specifically in Blockchain). This cost depends on the DLT type, platform, and programmed smart contract.

In evaluating the *DLT type*, for the permissioned types, the transaction cost can be defined based on the agreement of the involved parties. This means that in private/consortium DLTs, the enterprise may decide not to charge a sender for transmitting transactions. Indeed, for permissionless DLTs, transaction costs are mandatory. It can reduce the propagation of spam in the network and encourages miners/validators to contribute to the network's security. From the perspective of *DLT platform*, for different DLTs, the transaction cost is calculated differently. Finally, in the case of smart contracts, the transaction cost is highly dependent on the implementation. If the deployed smart contract requires many transactions to authenticate or authorize a user and creates many logs to allow a user to access the system, its cost would be very high in a public DLT. Therefore, choosing a suitable DLT and optimizing the smart contracts would significantly decrease transaction costs.

9. Future directions

The systematic examination of the existing DLT-based AAC methods has offered several insights into their challenges, obstacles, and benefits. It also leads us to propose several future directions to stimulate research efforts in this area. Several suggestions for future avenues are provided in two main categories, as follows: 1) the recommendations for improving the DLT-based AAC methods; the main target of these challenges is the *Authentication, Authorization, and Accountability (AAA)* research community. 2) the suggestions to improve/optimize DLT for AAC use cases; the main audience of these directions is the *DLT* research communities. A summary of the proposed future directions is depicted in Fig. 5.

9.1. Future directions to improve DLT-based AAC solutions

9.1.1. Security related improvements

Several proposals for security improvements are discussed below:

- *Privacy preserving*: The following two categories of privacy violation can be observed in existing methods: *user privacy (i.e., PII) violation, and policy/ rule privacy violation*. Targeting privacy preservation in these categories would be a promising and challenging future direction. Several methods can be beneficial in this regard: 1) hiding the policies and attributes of the AAC method using CP-ABE [191]; 2) using self-sovereign identities, and 3) benefiting from the modern cryptographic techniques to preserve user/rule privacy [25]. These solutions need to be optimized in terms of storage and latency.

- *Maintainability*: As a vital requirement for managing upcoming problems and providing adaptable systems, how to sustain maintainability is an ongoing problem. Once smart contracts have been published in a Blockchain, there is no means to change them. Providing flexible smart contracts using certain modifications (e.g., removing any constant addresses) [209] can mitigate this problem.

- *Supporting mutual authentication*: Thanks to the public-key infrastructure of DLTs, providing mutual authentication, to avoid several well-known attacks such as MitM and Reply, is more feasible and applicable compared to conventional methods.

- *Benefiting from more secure authentication types*: Most of the existing systems operate based on possession-based or knowledge-based one-factor authentication. The biometric and multi-factor authentications accompanying DLT's security features make the system more secure and reliable. It is worth mentioning that system optimization, decreasing transaction time and increasing scalability are challenging factors in this regard.

- *Removing the inherited vulnerabilities of centralized systems*: Although the proposed methods use DLT to improve the system functionality from different perspectives, having central entities is their weak point. Some of the existing solutions have central off-chain databases, a central authority, and management points that can be single points of failure and can increase the maintenance complexity of the systems. Increasing the automated part of the AAC procedure (e.g., access control decision-making) can help to resolve or mitigate these kinds of problems. The combination of cloud computing and DLT can be another practical scenario.

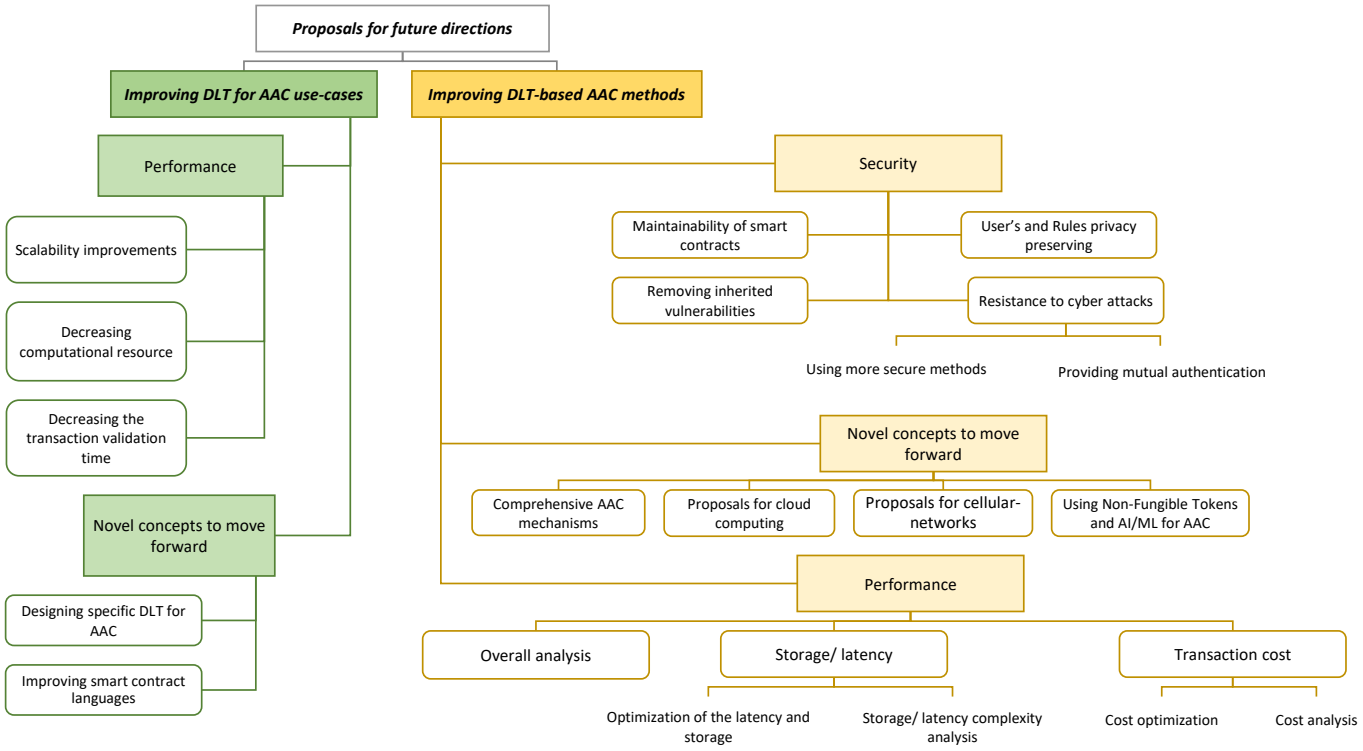

Figure 5: Proposals for future directions.

9.1.2. *Performance-related improvement*

The performance of existing systems can be enhanced in terms of the following aspects:

1. *Cost analysis*: The implementation cost of a DLT-based AAC system and the incentives for different parties to implement these methods are not clear in most of the proposed methods. One obstacle to migration to DLT-based AAC methods could be the lack of clearance from the investors. Therefore, we believe a comprehensive analysis in this regard would produce valuable insights that could encourage enterprises to enter this type of business.

2. *Optimization of transaction cost*: Each transaction making an update in the state of the DLT ledger can charge the sender. This cost varies according to the DLT types. For instance, in permissioned Blockchain, there is no mandatory charge for updating the ledger, and the transactions can be free of cost. Sophisticated programming to make minimum changes and automatically update the ledger can decrease the operation cost of the system. Moreover, a comprehensive analysis of different types of DLTs makes it possible to choose the most effective solution.

3. *Storage and latency*: Storage complexity and system latency are highly dependent on the implementation scenario. Thus, proposing efficient solutions for storage/latency optimization and security improvements would be a rewarding future direction. Some proposals in this regard could be the utilization of Oracles, the integration of conventional systems and DLT, and using cloud systems in the AAC procedure.

4. *Comprehensive analysis*: One of the obstacles to the proposed systems is the lack of transparency in implementation and analysis. Security analyses of well-known attacks and assessments of the performance, throughput, and scalability of the proposed systems would be an effective way to ease their implementation.

9.1.3. *Novel concepts to move forward*

Several novel concepts could be the subject of future studies:

1. *Benefiting from Machine Learning and Artificial Intelligence (ML/AI):* There are several challenges in DLT and the consensus procedure that can defect the performance of AAC methods based on this technology. For instance, high consensus convergence time and storage requirement. Due to [210], machine learning algorithms can effectively help to build more helpful and informative chains, and increase the data sharing speed due to the computational power of ML. So, using these technologies (i.e., AI/ML) to improve the DLT-based AAC mechanisms and address the existing challenges of DLT and smart contracts can be an interesting issue to go toward.

2. *Comprehensive AAC mechanisms* are proposed by a limited number of works. Providing a complete AAC mechanism and an Identity and Access Management (IAM) system are attractive fields of research.

3. *Benefiting from Non-Fungible Tokens (NFTs):* NFTs are cryptography assets, bound to their owner, and they are not interchangeable, mostly known as ERC721 tokens in Ethereum Blockchain. This technology represents ownership over digital or physical assets. Based on [211], NFTs offer verifiable immutability and authenticity, fast and convenient delegation, transfer of ownership, and revocation. Moreover, due to [212], this technology is highly beneficial regarding data sharing, and identity management.

4. *New concepts in several use-cases:* There are some pristine research subjects for AAC in different use cases. For example, several DLT-based AAC solutions have been proposed for resource/data sharing, and usage surveillance [12] in cloud computing. To the best of our knowledge, Virtual Machine (VM) access management remained intact. Using DLT in VM access management can improve a system's resilience against side-channel attacks [12]. Moreover, in cellular networks, user connectivity management, authentication, and access control could be motivating areas for network providers.

9.2. Future direction to improve DLT for AAC use-cases

The intrinsic limitations of DLT can slow down the flourishing process of DLT-based AAC solutions. So, in this section we propose several future directions for DLT community researchers to improve the existing DLTs to make them more suitable for AAC solutions.

9.2.1. Performance-related improvement

1. *Scalability*: The scalability problem of DLT in some platforms is a serious obstacle for AAC methods. The main goal of increasing the scalability is to provide higher throughput while growing the number of concurrent transactions. Generally, there are two main dimensions of Blockchain scalability, namely *horizontal* and *vertical*. Horizontal scalability refers to the capability of Blockchain to increase the throughput (or at least not to degrade it) by adding new nodes, while vertical scalability aims to enhance the capabilities of participating nodes to achieve higher throughput. On one hand, while public DLTs are more scalable in terms of the number of users, using these types can decrease the system throughput, increase the cost, and cause delays. Due to these problems, using permissioned DLTs seems to be a promising alternative. On the other hand, permissioned DLTs can threaten the user's privacy and anonymity. So, proposing a scalable tailor-made DLT dedicated to AAC would be an interesting subject for future work.

2. *Computational resources optimization*: One of the vital obstacles and challenging issues in using DLT for AAC is the resource consumption of the existing consensus methods. Most of the platforms have high resource consumption or require specific hardware. So, for the resource constraint devices, it is not feasible to benefit from the advantages of DLT. New studies could propose new models that have low resource consumption.

3. *Overall performance improvement*: One of the requirements of AAC systems is to provide rapid validation and verification. Existing consensus mechanisms take almost unacceptable verification times. High latency in the consensus algorithms and low block size can be the main causes of this problem. Proposing high throughput, and rapid algorithms will be a huge step forward.

9.2.2. Novel concepts to move forward

1. *Improving smart contract languages to support policy definition*: Existing smart contract languages such as Solidity, are designed for general-purpose use cases, so, they don't support the syntax and rules of existing well-known policy definition languages. Indeed, improving the smart contract languages to support well-known syntaxes such as eXtensible Access Control Markup Language (XACML) and XML Access Control Language.

2. *Designing comprehensive solution*: As a generic proposal for the DLT community, designing an AAC-specific DLT can be a significant step forward to provide all system requirements (e.g., scalability, minimum delay, user privacy, and low resource consumption). To sum up, we prompt the DLT and AAA research communities to work together on these challenges, enabling large-scale AAC deployments for networking applications.

References

[1] J. Liu, Y. Xiao, C. P. Chen, Authentication and access control in the internet of things, in: 2012 32nd International Conference on Distributed Computing Systems Workshops, IEEE, 2012, pp. 588–592.

[2] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, Guide to Attribute Based Access Control (ABAC) Definition and Considerations, Tech. Rep. NIST SP 800-162, National Institute of Standards and Technology (Jan. 2014). doi:10.6028/NIST.SP.800-162.

[3] A. Gauhar, N. Ahmad, Y. Cao, S. Khan, H. Cruickshank, E. A. Qazi, A. Ali, xDBAuth: Blockchain based cross domain authentication and authorization framework for Internet of Things, IEEE Access 8 (2020) 58800–58816, publisher: IEEE.

[4] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System 9.

[5] D. G. Wood, ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER 39.

[6] Secure Property Titles with Owner Authority | Satoshi Nakamoto Institute.
URL `https://nakamotoinstitute.org/secure-property-titles/`

[7] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, S. W. Kim, The future of healthcare internet of things: a survey of emerging technologies, IEEE Communications Surveys & Tutorials 22 (2) (2020) 1121–1167, publisher: IEEE.

[8] S. Sun, R. Du, S. Chen, W. Li, Blockchain-based iot access control system: Towards security, lightweight, and cross-domain, IEEE Access 9 (2021) 36868–36878. doi:10.1109/ACCESS.2021.3059863.

[9] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, Y. Liu, A survey of blockchain technology applied to smart cities: Research issues and challenges, IEEE Communications Surveys & Tutorials 21 (3) (2019) 2794–2830, publisher: IEEE.

[10] S. Ding, M. Ma, An attribute-based access control mechanism for blockchain-enabled internet of vehicles, in: Advances in Computer, Communication and Computational Sciences, Springer, 2021, pp. 905–915.

[11] D. C. Nguyen, P. N. Pathirana, M. Ding, A. Seneviratne, Integration of blockchain and cloud of things: Architecture, applications and challenges, IEEE Communications Surveys & Tutorials 22 (4) (2020) 2521–2549, publisher: IEEE.

[12] K. Gai, J. Guo, L. Zhu, S. Yu, Blockchain meets cloud computing: a survey, IEEE Communications Surveys & Tutorials 22 (3) (2020) 2009–2030, publisher: IEEE.

[13] R. Yang, F. R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: A survey, some research issues and challenges, IEEE Communications Surveys & Tutorials 21 (2) (2019) 1508–1532, publisher: IEEE.

[14] H. Baniata, A. Kertesz, A survey on blockchain-fog integration approaches, IEEE Access 8 (2020) 102657–102668, publisher: IEEE.

[15] G. Aceto, V. Persico, A. Pescape, A survey on information and communication technologies for Industry 4.0: state-of-the-art, taxonomies, perspectives, and challenges, IEEE Communications Surveys & Tutorials 21 (4) (2019) 3467–3501, publisher: IEEE.

[16] N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, P. N. Pathirana, A survey on blockchain for big data: approaches, opportunities, and future directions, Future Generation Computer Systems (2022).

[17] W. Gao, W. G. Hatcher, W. Yu, A survey of blockchain: Techniques, applications, and challenges, in: 2018 27th international conference on computer communication and networks (ICCCN), IEEE, 2018, pp. 1–11.

[18] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: A survey, International Journal of Web and Grid Services 14 (4) (2018) 352–375, publisher: Inderscience Publishers (IEL).

[19] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, Y. Yang, A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling, ACM Computing Surveys (CSUR) 53 (1) (2020) 1–32, publisher: ACM New York, NY, USA.

[20] S. Saxena, B. Bhushan, M. A. Ahad, Blockchain based solutions to secure iot: Background, integration trends and a way forward, Journal of Network and Computer Applications 181 (2021) 103050. doi:https://doi.org/10.1016/j.jnca.2021.103050.

[21] L. Da Xu, Y. Lu, L. Li, Embedding blockchain technology into iot for security: a survey, IEEE Internet of Things Journal (2021).

[22] A. H. Lone, R. Naaz, Applicability of blockchain smart contracts in securing internet and iot: a systematic literature review, Computer Science Review 39 (2021) 100360.

[23] S. Pal, A. Dorri, R. Jurdak, Blockchain for iot access control: Recent trends and future research directions, Journal of Network and Computer Applications (2022) 103371.

[24] T. Rathod, N. K. Jadav, M. D. Alshehri, S. Tanwar, R. Sharma, R.-A. Felseghi, M. S. Raboaca, Blockchain for future wireless networks: A decade survey, Sensors 22 (11) (2022) 4182.

[25] K. Yue, Y. Zhang, Y. Chen, Y. Li, L. Zhao, C. Rong, L. Chen, A survey of decentralizing applications via blockchain: The 5g and beyond perspective, IEEE Communications Surveys & Tutorials (2021).

[26] X. S. Shen, D. Liu, C. Huang, L. Xue, H. Yin, W. Zhuang, R. Sun, B. Ying, Blockchain for transparent data management toward 6g, Engineering 8 (2022) 74–85.

[27] A. J. Perez, S. Zeadally, Secure and privacy-preserving crowdsensing using smart contracts: Issues and solutions, Computer Science Review 43 (2022) 100450. doi:https://doi.org/10.1016/j.cosrev.2021.100450.

[28] T. Salman, M. Zolanvari, A. Erbad, R. Jain, M. Samaka, Security services using blockchains: A state of the art survey, IEEE Communications Surveys & Tutorials 21 (1) (2018) 858–880, publisher: IEEE.

[29] S. Y. Lim, P. T. Fotsing, A. Almasri, O. Musa, M. L. M. Kiah, T. F. Ang, R. Ismail, Blockchain technology the identity management and authentication service disruptor: a survey, Int. J. Adv. Sci. Eng. Inf. Technol 8 (4-2) (2018) 1735–1745.

[30] P. Mundhe, S. Verma, S. Venkatesan, A comprehensive survey on authentication and privacy-preserving schemes in vanets, Computer Science Review 41 (2021) 100411.

[31] F. Ghaffari, E. Bertin, J. Hatin, N. Crespi, Authentication and Access Control based on Distributed Ledger Technology: A survey, in: 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), IEEE, 2020, pp. 79–86.

[32] F. Ghaffari, K. Gilani, E. Bertin, N. Crespi, Identity and access management using distributed ledger technology: A survey, International Journal of Network Management (2021) e2180.

[33] D. Berdik, S. Otoum, N. Schmidt, D. Porter, Y. Jararweh, A survey on blockchain for information systems management and security, Information Processing & Management 58 (1) (2021) 102397. doi:https://doi.org/10.1016/j.ipm.2020.102397.

[34] O. Hasan, L. Brunie, E. Bertino, Privacy-preserving reputation systems based on blockchain and other cryptographic building blocks: A survey, ACM Computing Surveys (CSUR) 55 (2) (2022) 1–37.

[35] F. H. Pohrmen, R. K. Das, G. Saha, Blockchain-based security aspects in heterogeneous Internet-of-Things networks: A survey, Transactions on Emerging Telecommunications Technologies 30 (10) (2019) e3741, publisher: Wiley Online Library.

[36] B. Kitchenham, Procedures for undertaking systematic reviews: Joint technical report, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd.(0400011T. 1) (2004).

[37] G. Lame, Systematic literature reviews: An introduction, in: proceedings of the design society: international conference on engineering design, Vol. 1, Cambridge University Press, 2019, pp. 1633–1642.

[38] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering–a systematic literature review, Information and software technology 51 (1) (2009) 7–15.

[39] Y. Xiao, N. Zhang, W. Lou, Y. T. Hou, A Survey of Distributed Consensus Protocols for Blockchain Networks, IEEE Communications Surveys Tutorials 22 (2) (2020) 1432–1465, conference Name: IEEE Communications Surveys Tutorials. doi:10.1109/COMST.2020.2969706.

[40] N. Kannengießer, S. Lins, T. Dehling, A. Sunyaev, Trade-offs between distributed ledger technology characteristics, ACM Computing Surveys (CSUR) 53 (2) (2020) 1–37, publisher: ACM New York, NY, USA.

[41] DagCoin: a cryptocurrency without blocks (Sep. 2015).
URL `https://bitslog.com/2015/09/11/dagcoin/`

[42] Y. Sompolinsky, Y. Lewenberg, A. Zohar, Spectre: a fast and scalable cryptocurrency protocol., IACR Cryptol. ePrint Arch. 2016 (1159) (2016).

[43] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. M. Kayes, M. Alazab, P. Watters, A Comparative Analysis of Distributed Ledger Technology Platforms, IEEE Access 7 (2019) 167930–167943, conference Name: IEEE Access. doi:10.1109/ACCESS.2019.2953729.

[44] M. Li, C. Lal, M. Conti, D. Hu, Lechain: A blockchain-based lawful evidence management scheme for digital forensics, Future Generation Computer Systems 115 (2021) 406–420. doi:https://doi.org/10.1016/j.future.2020.09.038.

[45] G. Kumar, R. Saha, C. Lal, M. Conti, Internet-of-forensic (iof): A blockchain based digital forensics framework for iot applications, Future Generation Computer Systems 120 (2021) 13–25. doi:https://doi.org/10.1016/j.future.2021.02.016.

[46] I. Homoliak, S. Venugopalan, D. Reijsbergen, Q. Hum, R. Schumi, P. Szalachowski, The Security Reference Architecture for Blockchains: Toward a Standardized Model for Studying Vulnerabilities, Threats, and Defenses, IEEE Communications Surveys Tutorials 23 (1) (2021) 341–390, conference Name: IEEE Communications Surveys Tutorials. doi:10.1109/COMST.2020.3033665.

[47] L. Lamport, et al., Paxos made simple, ACM Sigact News 32 (4) (2001) 18–25.

[48] L. Ismail, H. Materwala, A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions, Symmetry 11 (10) (2019) 1198, publisher: Multidisciplinary Digital Publishing Institute.

[49] C. Dwork, M. Naor, Pricing via processing or combatting junk mail, in: Annual international cryptology conference, Springer, 1992, pp. 139–147.

[50] M. Jakobsson, A. Juels, Proofs of work and bread pudding protocols, in: Secure information networks, Springer, 1999, pp. 258–272.

[51] S. King, S. Nadal, PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake 6.

[52] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, PBFT vs Proof-of-Authority: Applying the CAP Theorem to Permissioned Blockchain 11.

[53] K. Salah, M. H. U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for AI: Review and open research challenges, IEEE Access 7 (2019)

10127–10149, publisher: IEEE.
[54] Delegated Proof of Stake (DPOS) — BitShares Documentation documentation.
URL https://how.bitshares.works/en/master/technology/dpos.html
[55] M. Castro, B. Liskov, Practical byzantine fault tolerance, in: OSDI, Vol. 99, 1999, pp. 173–186, issue: 1999.
[56] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 USENIX Annual Technical Conference (USENIX ATC 14), 2014, pp. 305–319.
[57] A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in: Annual International Cryptology Conference, Springer, 2017, pp. 357–388.
[58] Y. Wang, S. Cai, C. Lin, Z. Chen, T. Wang, Z. Gao, C. Zhou, Study of blockchains's consensus mechanism based on credit, IEEE Access 7 (2019) 10224–10231, publisher: IEEE.
[59] Y. Liu, F. R. Yu, X. Li, H. Ji, V. C. Leung, Blockchain and machine learning for communications and networking systems, IEEE Communications Surveys & Tutorials 22 (2) (2020) 1392–1431, publisher: IEEE.
[60] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, D. Mohaisen, Exploring the Attack Surface of Blockchain: A Comprehensive Survey, IEEE Communications Surveys Tutorials 22 (3) (2020) 1977–2008, conference Name: IEEE Communications Surveys Tutorials. doi:10.1109/COMST.2020.2975999.
[61] A. K. Jain, A. Ross, S. Prabhakar, An introduction to biometric recognition, IEEE Transactions on circuits and systems for video technology 14 (1) (2004) 4–20, publisher: IEEE.
[62] D. Ferraiolo, D. R. Kuhn, R. Chandramouli, Role-based access control, Artech House, 2003.
[63] B. W. Lampson, Protection, ACM SIGOPS Operating Systems Review 8 (1) (1974) 18–24.
[64] H. M. Levy, Capability-based computer systems, Digital Press, 2014.
URL https://homes.cs.washington.edu/ levy/capabook/
[65] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, B. Fang, A Survey on Access Control in the Age of Internet of Things, IEEE Internet of Things Journal 7 (6) (2020) 4682–4696, conference Name: IEEE Internet of Things Journal. doi:10.1109/JIOT.2020.2969326.
[66] P. Samarati, S. C. de Vimercati, Access control: Policies, models, and mechanisms, in: International School on Foundations of Security Analysis and Design, Springer, 2000, pp. 137–196.
[67] R. S. Sandhu, Role-based access control, in: Advances in computers, Vol. 46, Elsevier, 1998, pp. 237–286.
[68] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 89–98.
[69] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: International Workshop on Public Key Cryptography, Springer, 2011, pp. 53–70.
[70] X. Fu, Y. Ding, H. Li, J. Ning, T. Wu, F. Li, A survey of lattice based expressive attribute based encryption, Computer Science Review 43 (2022) 100438.
[71] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: 2007 IEEE symposium on security and privacy (SP'07), IEEE, 2007, pp. 321–334.
[72] B. Sumitra, C. R. Pethuru, M. Misbahuddin, A Survey of Cloud Authentication Attacks and Solution Approaches, in: International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), An ISO 3297:2007, ISSN (online): 23209801, ISSN (Print): 2320-9798, Volume 2, Issue 10, 2014.
[73] C. Lin, D. He, X. Huang, K.-K. R. Choo, A. V. Vasilakos, BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0, Journal of Network and Computer Applications 116 (2018) 42–52, publisher: Elsevier.
[74] L. Zhang, H. Li, L. Sun, Z. Shi, Y. He, Poster: towards fully distributed user authentication with blockchain, in: 2017 IEEE Symposium on Privacy-Aware Computing (PAC), IEEE, 2017, pp. 202–203.
[75] G. Deep, R. Mohana, A. Nayyar, P. Sanjeevikumar, E. Hossain, Authentication protocol for cloud databases using blockchain mechanism, Sensors 19 (20) (2019) 4444, publisher: Multidisciplinary Digital Publishing Institute.
[76] J. R. Douceur, The sybil attack, in: International workshop on peer-to-peer systems, Springer, 2002, pp. 251–260.
[77] O. Delgado-Mohatar, J. Fierrez, R. Tolosana, R. Vera-Rodriguez, Blockchain and biometrics: A first look into opportunities and challenges, in: International Congress on Blockchain and Applications, Springer, 2019, pp. 169–177.
[78] N. Abdullah, A. Hakansson, E. Moradian, Blockchain based approach to enhance big data authentication in distributed environment, in: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2017, pp. 887–892.
[79] ION – Booting up the network, section: Identity Standards Blog (Jun. 2020).
URL https://techcommunity.microsoft.com/t5/identity-standards-blog/ion-booting-up-the-network/ba-p/1441552
[80] Identity protocol v1 - Bitcoin Wiki.
URL https://en.bitcoin.it/wiki/Identity_protocol_v1
[81] A. Mühle, A. Grüner, T. Gayvoronskaya, C. Meinel, A survey on essential components of a self-sovereign identity, Computer Science Review 30 (2018) 80–86.
[82] K. Baqer, D. Y. Huang, D. McCoy, N. Weaver, Stressing out: Bitcoin "stress testing", in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 3–18.
[83] Z. Cui, X. Fei, S. Zhang, X. Cai, Y. Cao, W. Zhang, J. Chen, A hybrid blockchain-based identity authentication scheme for multi-wsn, IEEE Transactions on Services Computing 13 (2) (2020) 241–251.
[84] L. Xiong, F. Li, S. Zeng, T. Peng, Z. Liu, A blockchain-based privacy-awareness authentication scheme with efficient revocation for multi-server architectures, IEEE Access 7 (2019) 125840–125853.
[85] M. A. Khan, K. Salah, IoT security: Review, blockchain solutions, and open challenges, Future Generation Computer Systems 82 (2018) 395–411. doi:10.1016/j.future.2017.11.022.
[86] K. Bicakci, D. Unal, N. Ascioglu, O. Adalier, Mobile authentication secure against man-in-the-middle attacks, Procedia Computer Science 34 (2014) 323–329.
[87] A. Z. Ourad, B. Belgacem, K. Salah, Using blockchain for iot access control and authentication management, in: International Conference on Internet of Things, Springer, 2018, pp. 150–164.
[88] A. Yakubov, W. Shbair, N. Khan, C. Medinger, J. Hilger, et al., Blockpgp: A blockchain-based framework for pgp key servers, International Journal of Networking and Computing 10 (1) (2020) 1–24.
[89] B. Alotaibi, Utilizing blockchain to overcome cyber security concerns in the internet of things: A review, IEEE Sensors Journal 19 (23) (2019) 10953–10971, publisher: IEEE.
[90] R. John, J. P. Cherian, J. J. Kizhakkethottam, A survey of techniques to prevent sybil attacks, in: 2015 International Conference on Soft-Computing and Networks Security (ICSNS), 2015, pp. 1–6. doi:10.1109/ICSNS.2015.7292385.
[91] M. Conti, E. S. Kumar, C. Lal, S. Ruj, A Survey on Security and Privacy Issues of Bitcoin, IEEE Communications Surveys Tutorials 20 (4) (2018) 3416–3452, conference Name: IEEE Communications Surveys Tutorials. doi:10.1109/COMST.2018.2842460.
[92] K. Alachkar, D. Gaastra, Blockchain-based Sybil Attack Mitigation: A Case Study of the I2P Network, Semanticscholar Seattle, Washington, 2018.
[93] M. Belotti, N. Božić, G. Pujolle, S. Secci, A vademecum on blockchain technologies: When, which, and how, IEEE Communications Surveys & Tutorials 21 (4) (2019) 3796–3838, publisher: IEEE.
[94] A. Panarello, N. Tapas, G. Merlino, F. Longo, A. Puliafito, Blockchain and iot integration: A systematic survey, Sensors 18 (8) (2018) 2575.
[95] P. M. Mell, T. Grance, Sp 800-145. the nist definition of cloud computing (2011).
URL https://dl.acm.org/doi/pdf/10.5555/2206223
[96] N. Abramson, The aloha system: Another alternative for computer communications, in: Proceedings of the November 17-19, 1970, fall joint computer conference, 1970, pp. 281–285.
[97] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, A survey of information-centric networking research, IEEE communications surveys & tutorials 16 (2) (2013) 1024–1049.
[98] M. Conti, M. Hassan, C. Lal, Blockauth: Blockchain based distributed producer authentication in icn, Computer Networks 164 (2019) 106888.

[99] J.-H. Huh, K. Seo, Blockchain-based mobile fingerprint verification and automatic log-in platform for future computing, The Journal of Supercomputing 75 (6) (2019) 3123–3139.
[100] L. Wu, X. Du, W. Wang, B. Lin, An out-of-band authentication scheme for internet of things using blockchain technology, in: 2018 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2018, pp. 769–773.
[101] M. T. Hammi, B. Hammi, P. Bellot, A. Serhrouchni, Bubbles of trust: A decentralized blockchain-based authentication system for iot, Computers & Security 78 (2018) 126–142.
[102] J.-H. Lee, Bidaas: Blockchain based id as a service, IEEE Access 6 (2017) 2274–2278.
[103] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, Blockchain-enabled authentication handover with efficient privacy protection in sdn-based 5g networks, IEEE Transactions on Network Science and Engineering (2019).
[104] Y. Zhang, R. Deng, E. Bertino, D. Zheng, Robust and universal seamless handover authentication in 5g hetnets, IEEE Transactions on Dependable and Secure Computing (2019).
[105] K. Xue, X. Luo, Y. Ma, J. Li, J. Liu, D. S. Wei, A distributed authentication scheme based on smart contract for roaming service in mobile vehicular networks, IEEE Transactions on Vehicular Technology (2022).
[106] H. Lee, M. Ma, Blockchain-based mobility management for 5g, Future Generation Computer Systems 110 (2020) 638–646.
[107] T. Sanda, H. Inaba, Proposal of new authentication method in wi-fi access using bitcoin 2.0, in: 2016 IEEE 5th Global Conference on Consumer Electronics, IEEE, 2016, pp. 1–5.
[108] Y. Niu, L. Wei, C. Zhang, J. Liu, Y. Fang, An anonymous and accountable authentication scheme for wi-fi hotspot access with the bitcoin blockchain, in: 2017 IEEE/CIC International Conference on Communications in China (ICCC), IEEE, 2017, pp. 1–6.
[109] A. Mohsin, A. Zaidan, B. Zaidan, O. Albahri, A. Albahri, M. Alsalem, K. Mohammed, Based blockchain-pso-aes techniques in finger vein biometrics: A novel verification secure framework for patient authentication, Computer Standards & Interfaces 66 (2019) 103343.
[110] M. T. Hammi, P. Bellot, A. Serhrouchni, Bctrust: A decentralized authentication blockchain-based mechanism, in: 2018 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2018, pp. 1–6.
[111] A. Moinet, B. Darties, J.-L. Baril, Blockchain based trust & authentication for decentralized sensor networks, arXiv preprint arXiv:1706.01730 (2017).
URL https://arxiv.org/pdf/1706.01730.pdf
[112] H.-W. Kim, Y.-S. Jeong, Secure authentication-management human-centric scheme for trusting personal resource information on mobile cloud computing with blockchain, Human-centric Computing and Information Sciences 8 (1) (2018) 1–13.
[113] M. Wazid, A. K. Das, R. Hussain, N. Kumar, S. Roy, Buaka-cs: Blockchain-enabled user authentication and key agreement scheme for crowdsourcing system, Journal of Systems Architecture 123 (2022) 102370.
[114] B. Leiding, C. H. Cap, T. Mundt, S. Rashidibajgan, Authcoin: validation and authentication in decentralized networks, arXiv preprint arXiv:1609.04955 (2016).
URL https://arxiv.org/ftp/arxiv/papers/1609/1609.04955.pdf
[115] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, X. Lin, Ptas: Privacy-preserving thin-client authentication scheme in blockchain-based pki, Future Generation Computer Systems 96 (2019) 185–195.
[116] C. Fromknecht, D. Velicanu, CertCoin : A NameCoin Based Decentralized Authentication System (2014).
URL https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf
[117] G. Ali, N. Ahmad, Y. Cao, M. Asif, H. Cruickshank, Q. E. Ali, Blockchain based permission delegation and access control in internet of things (baci), Computers & Security 86 (2019) 318–334.
[118] R. Xu, Y. Chen, E. Blasch, G. Chen, Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot, Computers 7 (3) (2018) 39.
[119] Y. Nakamura, Y. Zhang, M. Sasabe, S. Kasahara, Exploiting smart contracts for capability-based access control in the internet of things, Sensors 20 (6) (2020) 1793.
[120] N. Tapas, F. Longo, G. Merlino, A. Puliafito, Experimenting with smart contracts for access control and delegation in iot, Future Generation Computer Systems 111 (2020) 324–338.
[121] F. Longo, D. Bruneo, S. Distefano, G. Merlino, A. Puliafito, Stack4things: a sensing-and-actuation-as-a-service framework for iot and cloud integration, Annals of Telecommunications 72 (1-2) (2017) 53–70.
[122] T. Le, M. W. Mutka, Capchain: A privacy preserving access control framework based on blockchain for pervasive environments, in: 2018 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2018, pp. 57–64.
[123] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, Y. Zhang, Blockchain for secure and efficient data sharing in vehicular edge computing and networks, IEEE Internet of Things Journal 6 (3) (2019) 4660–4670. doi:10.1109/JIOT.2018.2875542.
[124] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, N. Javaid, Data sharing system integrating access control mechanism using blockchain-based smart contracts for iot devices, Applied Sciences 10 (2) (2020) 488.
[125] T. Sultana, A. Ghaffar, M. Azeem, Z. Abubaker, M. U. Gurmani, N. Javaid, Data sharing system integrating access control based on smart contracts for iot, in: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Springer, 2019, pp. 863–874.
[126] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, J. Wan, Smart contract-based access control for the internet of things, IEEE Internet of Things Journal 6 (2) (2018) 1594–1605.
[127] H. Shafagh, L. Burkhalter, A. Hithnawi, S. Duquennoy, Towards blockchain-based auditable storage and sharing of iot data, in: Proceedings of the 2017 on Cloud Computing Security Workshop, 2017, pp. 45–50.
[128] S. Dramé-Maigné, M. Laurent, L. Castillo, Distributed access control solution for the iot based on multi-endorsed attributes and smart contracts, in: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2019, pp. 1582–1587.
[129] O. J. A. Pinno, A. R. A. Gregio, L. C. De Bona, Controlchain: Blockchain as a central enabler for access control authorizations in the iot, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–6.
[130] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, A. Ignjatovic, Trust-based blockchain authorization for iot, IEEE Transactions on Network and Service Management (2021).
[131] Y. Zhang, B. Li, B. Liu, J. Wu, Y. Wang, X. Yang, An attribute-based collaborative access control scheme using blockchain for iot devices, Electronics 9 (2) (2020) 285.
[132] H. Liu, D. Han, D. Li, Fabric-iot: A blockchain-based access control system in iot, IEEE Access 8 (2020) 18207–18218.
[133] S. Ding, J. Cao, C. Li, K. Fan, H. Li, A novel attribute-based access control scheme using blockchain for iot, IEEE Access 7 (2019) 38431–38441.
[134] M. A. Islam, S. Madria, A permissioned blockchain based access control system for iot, in: 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, 2019, pp. 469–476.
[135] M. Yutaka, Y. Zhang, M. Sasabe, S. Kasahara, Using ethereum blockchain for distributed attribute-based access control in the internet of things, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[136] S. Figueroa, J. Añorga, S. Arrizabalaga, An attribute-based access control model in rfid systems based on blockchain decentralized applications for healthcare environments, Computers 8 (3) (2019) 57.
[137] A. Dorri, S. S. Kanhere, R. Jurdak, P. Gauravaram, Lsb: A lightweight scalable blockchain for iot security and anonymity, Journal of Parallel and Distributed Computing 134 (2019) 180–197.
[138] A. H. Mohsin, A. A. Zaidan, B. B. Zaidan, O. S. Albahri, A. S. Albahri, M. A. Alsalem, K. I. Mohammed, Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions, Computer Standards & Interfaces 64 (2019) 41–60, publisher: Elsevier.
[139] S. K. Lo, Y. Liu, S. Y. Chia, X. Xu, Q. Lu, L. Zhu, H. Ning, Analysis of blockchain solutions for iot: A systematic literature review, IEEE Access 7 (2019) 58822–58835.
[140] We (Finally) Built Eris!

URL https://blog.erisindustries.com/products/2015/08/19/v010-release/

[141] A. Menezes, P. Van Oorschot, S. Vanstone, Chapter 12 key establishment protocols, Handbook of Applied Cryptography (1997) 489–541.

[142] S. Banerjee, B. Bera, A. K. Das, S. Chattopadhyay, M. K. Khan, J. J. Rodrigues, Private blockchain-envisioned multi-authority cp-abe-based user access control scheme in iiot, Computer Communications 169 (2021) 99–113.

[143] H. Al Breiki, L. Al Qassem, K. Salah, M. H. U. Rehman, D. Sevtinovic, Decentralized access control for iot data using blockchain and trusted oracles, in: 2019 IEEE International Conference on Industrial Internet (ICII), IEEE, 2019, pp. 248–257.

[144] D. R. Putra, B. Anggorojati, A. P. P. Hartono, Blockchain and smart-contract for scalable access control in internet of things, in: 10th International Conference on ICT for Smart Society, ICISS 2019, Institute of Electrical and Electronics Engineers Inc., 2019, p. 8969807.

[145] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, F. Zanichelli, Iotchain: A blockchain security architecture for the internet of things, in: 2018 IEEE wireless communications and networking conference (WCNC), IEEE, 2018, pp. 1–6.

[146] O. Novo, Scalable access management in iot using blockchain: A performance evaluation, IEEE Internet of Things Journal 6 (3) (2018) 4694–4701.

[147] B. Tang, H. Kang, J. Fan, Q. Li, R. Sandhu, Iot passport: A blockchain-based trust framework for collaborative internet-of-things, in: Proceedings of the 24th ACM symposium on access control models and technologies, 2019, pp. 83–92.

[148] A. Outchakoucht, E. Hamza, J. P. Leroy, Dynamic access control policy based on blockchain and machine learning for the internet of things, Int. J. Adv. Comput. Sci. Appl 8 (7) (2017) 417–424.

[149] A. I. Abdi, F. E. Eassa, K. Jambi, K. Almarhabi, M. Khemakhem, A. Basuhail, M. Yamin, Hierarchical blockchain-based multi-chaincode access control for securing iot systems, Electronics 11 (5) (2022) 711.

[150] X. Hao, W. Ren, Y. Fei, T. Zhu, K.-K. R. Choo, A blockchain-based cross-domain and autonomous access control scheme for internet of things, IEEE Transactions on Services Computing (2022) 1–1doi:10.1109/TSC.2022.3179727.

[151] F. Ghaffari, E. Bertin, N. Crespi, S. Behrad, J. Hatin, A novel access control method via smart contracts for internet-based service provisioning, IEEE Access 9 (2021) 81253–81273. doi:10.1109/ACCESS.2021.3085831.

[152] F. Ghaffari, E. Bertin, N. Crespi, A novel approach for network resource sharing via blockchain, in: Proceedings of the SIGCOMM '21 Poster and Demo Sessions, SIGCOMM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 50–52. doi:10.1145/3472716.3472867.
URL https://doi.org/10.1145/3472716.3472867

[153] X. Ling, J. Wang, T. Bouchoucha, B. C. Levy, Z. Ding, Blockchain radio access network (b-ran): Towards decentralized secure radio access paradigm, IEEE Access 7 (2019) 9714–9723.

[154] Y. Le, X. Ling, J. Wang, Z. Ding, Prototype design and test of blockchain radio access network, in: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2019, pp. 1–6.

[155] X. Ling, Y. Le, J. Wang, Z. Ding, X. Gao, Practical modeling and analysis of blockchain radio access network, IEEE Transactions on Communications (2020).

[156] X. Ling, Y. Le, J. Wang, Z. Ding, Hash access: Trustworthy grant-free iot access enabled by blockchain radio access networks, IEEE Network 34 (1) (2020) 54–61.

[157] K. Fan, Y. Ren, Y. Wang, H. Li, Y. Yang, Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5g, IET communications 12 (5) (2018) 527–532.

[158] X. Qin, Y. Huang, Z. Yang, X. Li, An access control scheme with fine-grained time constrained attributes based on smart contract and trapdoor, in: 2019 26th International Conference on Telecommunications (ICT), IEEE, 2019, pp. 249–253.

[159] S. Alansari, F. Paci, V. Sassone, A distributed access control system for cloud federations, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2017, pp. 2131–2136.

[160] L. Guo, X. Yang, W.-C. Yau, Tabe-dac: Efficient traceable attribute-based encryption scheme with dynamic access control based on blockchain, IEEE Access 9 (2021) 8479–8490.

[161] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao, K. Yu, Authprivacychain: A blockchain-based access control framework with privacy protection in cloud, IEEE Access 8 (2020) 70604–70615.

[162] Y. Xiao, N. Zhang, J. Li, W. Lou, Y. T. Hou, Privacyguard: Enforcing private data usage control with blockchain and attested off-chain contract execution, in: European Symposium on Research in Computer Security, Springer, 2020, pp. 610–629.

[163] Y. Zhu, Y. Qin, G. Gan, Y. Shuai, W. C.-C. Chu, Tbac: transaction-based access control on blockchain for resource sharing with cryptographically decentralized authorization, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Vol. 1, IEEE, 2018, pp. 535–544.

[164] I. Sukhodolskiy, S. Zapechnikov, A blockchain-based access control system for cloud storage, in: 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), IEEE, 2018, pp. 1575–1578.

[165] G. Yu, X. Zha, X. Wang, W. Ni, K. Yu, P. Yu, J. A. Zhang, R. P. Liu, Y. J. Guo, Enabling attribute revocation for fine-grained access control in blockchain-iot systems, IEEE Transactions on Engineering Management 67 (4) (2020) 1213–1230.

[166] S. Wang, X. Wang, Y. Zhang, A secure cloud storage framework with access control based on blockchain, IEEE Access 7 (2019) 112713–112725.

[167] J. Zhang, Y. Yang, X. Liu, J. Ma, An efficient blockchain-based hierarchical data sharing for healthcare internet of things, IEEE Transactions on Industrial Informatics (2022).

[168] A. R. Rajput, Q. Li, M. T. Ahvanooey, I. Masood, Eacms: Emergency access control management system for personal health record based on blockchain, IEEE Access 7 (2019) 84304–84317.

[169] D. C. Nguyen, P. N. Pathirana, M. Ding, A. Seneviratne, Bedgehealth: A decentralized architecture for edge-based iomt networks using blockchain, IEEE Internet of Things Journal (2021).

[170] R. Li, T. Song, B. Mei, H. Li, X. Cheng, L. Sun, Blockchain for large-scale internet of things data storage and protection, IEEE Transactions on Services Computing 12 (5) (2018) 762–771.

[171] Q. Lyu, Y. Qi, X. Zhang, H. Liu, Q. Wang, N. Zheng, Sbac: A secure blockchain-based access control framework for information-centric networking, Journal of Network and Computer Applications 149 (2020) 102444.

[172] D. D. F. Maesa, P. Mori, L. Ricci, Blockchain based access control, in: IFIP international conference on distributed applications and interoperable systems, Springer, 2017, pp. 206–220.

[173] S. Shafeeq, M. Alam, A. Khan, Privacy aware decentralized access control system, Future Generation Computer Systems 101 (2019) 420–433.

[174] S. Popov, The tangle (2016).
URL https://iotatoken.com/IOTA_Whitepaper.pdf

[175] S. Fujimura, H. Watanabe, A. Nakadaira, T. Yamada, A. Akutsu, J. J. Kishigami, Bright: A concept for a decentralized rights management system based on blockchain, in: 2015 IEEE 5th International Conference on Consumer Electronics-Berlin (ICCE-Berlin), IEEE, 2015, pp. 345–346.

[176] C. Ihle, O. Sanchez, Smart contract-based role management on the blockchain, in: International Conference on Business Information Systems, Springer, 2018, pp. 335–343.

[177] G. Ali, N. Ahmad, Y. Cao, Q. E. Ali, F. Azim, H. Cruickshank, Bcon: Blockchain based access control across multiple conflict of interest domains, Journal of Network and Computer Applications 147 (2019) 102440.

[178] J. Paillisse, J. Subira, A. Lopez, A. Rodriguez-Natal, V. Ermagan, F. Maino, A. Cabellos, Distributed access control with blockchain, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), IEEE, 2019, pp. 1–6.

[179] S. Rouhani, R. Belchior, R. S. Cruz, R. Deters, Distributed attribute-based access control system using permissioned blockchain, World Wide Web (2021) 1–28.

[180] D. D. F. Maesa, P. Mori, L. Ricci, A blockchain based approach for the definition of auditable access control systems, Computers & Security 84 (2019) 93–119.

[181] D. D. F. Maesa, P. Mori, L. Ricci, Blockchain based access control ser-

vices, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2018, pp. 1379–1386.
[182] P. Wang, Y. Yue, W. Sun, J. Liu, An attribute-based distributed access control for blockchain-enabled iot, in: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2019, pp. 1–6.
[183] Y. Ding, J. Jin, J. Zhang, Z. Wu, K. Hu, Sc-rbac: A smart contract based rbac model for dapps, in: International Conference on Human Centered Computing, Springer, 2019, pp. 75–85.
[184] J. P. Cruz, Y. Kaji, N. Yanai, Rbac-sc: Role-based access control using smart contract, Ieee Access 6 (2018) 12240–12251.
[185] G. Zyskind, O. Nathan, A. Pentland, Enigma: Decentralized computation platform with guaranteed privacy, arXiv preprint arXiv:1506.03471 (2015).
[186] H. Shrobe, D. L. Shrier, A. Pentland, Enigma: Decentralized computation platform with guaranteed privacy (2018).
[187] G. Zyskind, O. Nathan, et al., Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops, IEEE, 2015, pp. 180–184.
[188] A. Kiran, S. Dharanikota, A. Basava, Blockchain based data access control using smart contracts, in: TENCON 2019-2019 IEEE Region 10 Conference (TENCON), IEEE, 2019, pp. 2335–2339.
[189] H. Bowen, L. Yi, F. Li, D. Xinhua, C. Ping, Blockchain-based Access Control Data Distribution System, in: 2019 IEEE 5th International Conference on Computer and Communications (ICCC), IEEE, 2019, pp. 1231–1236.
[190] A. Wu, Y. Zhang, X. Zheng, R. Guo, Q. Zhao, D. Zheng, Efficient and privacy-preserving traceable attribute-based encryption in blockchain, Annals of Telecommunications 74 (7) (2019) 401–411.
[191] S. Gao, G. Piao, J. Zhu, X. Ma, J. Ma, Trustaccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain, IEEE Transactions on Vehicular Technology 69 (6) (2020) 5784–5798.
[192] S. Wang, Y. Zhang, Y. Zhang, A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems, Ieee Access 6 (2018) 38437–38450.
[193] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, K. Salah, A user authentication scheme of iot devices using blockchain-enabled fog nodes, in: 2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA), IEEE, 2018, pp. 1–8.
[194] L. Widick, I. Ranasinghe, R. Dantu, S. Jonnada, Blockchain based authentication and authorization framework for remote collaboration systems, in: 2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), IEEE, 2019, pp. 1–7.
[195] A. Ouaddah, A. Abou Elkalam, A. Ait Ouahman, Fairaccess: a new blockchain-based access control framework for the internet of things, Security and communication networks 9 (18) (2016) 5943–5964.
[196] A. Ouaddah, A. Abou Elkalam, A. A. Ouahman, Towards a novel privacy-preserving access control model based on blockchain technology in iot, in: Europe and MENA cooperation advances in information and communication technologies, Springer, 2017, pp. 523–533.
[197] R. Akkaoui, X. Hei, W. Cheng, Edgemedichain: A hybrid edge blockchain-based framework for health data exchange, IEEE Access 8 (2020) 113467–113486.
[198] L. Babun, K. Denney, Z. B. Celik, P. McDaniel, A. S. Uluagac, A survey on iot platforms: Communication, security, and privacy perspectives, Computer Networks 192 (2021) 108040.
[199] C. Li, J. Zhang, X. Yang, L. Youlong, Lightweight blockchain consensus mechanism and storage optimization for resource-constrained iot devices, Information Processing & Management 58 (4) (2021) 102602.
[200] Y. Yin, Y. Li, B. Ye, T. Liang, Y. Li, A blockchain-based incremental update supported data storage system for intelligent vehicles, IEEE Transactions on Vehicular Technology (2021).
[201] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, X. Zhang, Bbds: Blockchain-based data sharing for electronic medical records in cloud environments, Information 8 (2) (2017) 44.
[202] S. Popov, W. J. Buchanan, Fpc-bi: Fast probabilistic consensus within byzantine infrastructures, Journal of Parallel and Distributed Computing 147 (2021) 77–86.
[203] J. Xue, C. Xu, Y. Zhang, Private blockchain-based secure access control for smart home systems, KSII Transactions on Internet and Information Systems (TIIS) 12 (12) (2018) 6057–6078.
[204] P. Gupta, V. Dedeoglu, K. Najeebullah, S. S. Kanhere, R. Jurdak, Energy-aware demand selection and allocation for real-time iot data trading, in: 2020 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2020, pp. 138–147.
[205] R. Uda, Vulnerable web server protection by hash based url transformation, in: 2020 54th Annual Conference on Information Sciences and Systems (CISS), IEEE, 2020, pp. 1–6.
[206] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, T. Chen, Defining smart contract defects on ethereum, IEEE Transactions on Software Engineering (2020) 1–1doi:10.1109/TSE.2020.2989002.
[207] M. A. Azad, S. Bag, F. Hao, A. Shalaginov, Decentralized self-enforcing trust management system for social internet of things, IEEE Internet of Things Journal 7 (4) (2020) 2690–2703.
[208] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, J. Son, Recent advances in smart contracts: A technical overview and state of the art, IEEE Access 8 (2020) 117782–117801.
[209] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, T. Chen, Defining smart contract defects on ethereum, IEEE Transactions on Software Engineering (2020).
[210] R. Gaur, S. Prakash, S. Kumar, K. Abhishek, M. Msahli, A. Wahid, A machine-learning–blockchain-based authentication using smart contracts for an ioht system, Sensors 22 (23) (2022) 9074.
[211] A. Sghaier Omar, O. Basir, Capability-based non-fungible tokens approach for a decentralized aaa framework in iot, in: Blockchain Cybersecurity, Trust and Privacy, Springer, 2020, pp. 7–31.
[212] Q. Wang, R. Li, Q. Wang, S. Chen, Non-fungible token (nft): Overview, evaluation, opportunities and challenges, arXiv preprint arXiv:2105.07447 (2021).