IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 July 2022, accepted 19 August 2022, date of publication 29 August 2022, date of current version 8 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202553

TOPICAL REVIEW

Comparative Analysis of Decentralized Identity Approaches

MORTEZA ALIZADEH, (Graduate Student Member, IEEE), KARL ANDERSSON, (Senior Member, IEEE), AND OLOV SCHELÉN, (Senior Member, IEEE)

Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 93187 Luleå, Sweden

Corresponding author: Morteza Alizadeh (morteza.alizadeh@ltu.se)

This work was supported by the Kolarctic CBC Project DIT4BEARS under Grant KO4096.

ABSTRACT Decentralization is essential when trust and performance must not depend on a single organization. Distributed Ledger Technologies (DLTs) and Decentralized Hash Tables (DHTs) are examples where the DLT is useful for transactional events, and the DHT is useful for large-scale data storage. The combination of these two technologies can meet many challenges. The blockchain is a DLT with immutable history protected by cryptographic signatures in data blocks. Identification is an essential issue traditionally provided by centralized trust anchors. Self-sovereign identities (SSIs) are proposed decentralized models where users can control and manage their identities with the help of DHT. However, slowness is a challenge among decentralized identification systems because of many connections and requests among participants. In this article, we focus on decentralized identification by DLT and DHT, where users can control their information and store biometrics. We survey some existing alternatives and address the performance challenge by comparing different decentralized identification technologies based on execution time and throughput. We show that the DHT and machine learning model (BioIPFS) performs better than other solutions such as uPort, ShoCard, and BBID.

INDEX TERMS Identification, decentralization, distributed hash table, self-sovereign identity.

I. INTRODUCTION

Decentralization is a primary feature of the Internet architecture. However, identifying entities in a decentralized system is still a research issue. Moreover, in recent years, identification systems have changed rapidly. For example, most digital devices, such as smartphones, need to identify their owners when communicating with applications or devices through the Internet. Centralized systems have operated most identification services to identify people or verify whom they claim to be. Typically, there are several security layers for different types of entities provided by centralized identification in one unified entry system. Moreover, many solutions today have policies that are managed centrally.

Cloud services provide remote resources such as data storage (cloud storage) and compute resources without direct active management by the user. Therefore, identification can be performed with the help of cloud services, where Azure, Google Cloud, and AWS are different clouds with different policies under Microsoft, Google, and Amazon management, respectively [1]. Although the cloud is an efficient solution, research exploring new solutions to improve the centralized sections of clouds has been performed [2].

A decentralized identification system should identify many users (a person, an organization, a thing, a data model, an abstract entity, etc.) without the help of any centralized entity. Furthermore, it must be able to recognize different identities with the help of other registered participants without any centralized registry, identity provider, or certificate authority to approve themselves for the whole system. Self-sovereign identity (SSI) is a decentralized technique for identification that provides individuals with control over their digital identities. These identification systems have unique, private, and secure peer-to-peer connections between parties [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Giuseppe Destefanis.

As the main module in SSI, decentralized identifiers help identify users who want to access decentralized applications. However, the main problem of the decentralized identifiers is that they are slow because of the need for many connections and requests among other network participants.

In this article, we look for faster solutions. Among other decentralized solutions, one of the efficient techniques is distributed hash table (DHT), which has quick lookup functions to load identifiers on key-value pairs [4]. DHT has several advantages, such as enabling scalability, setting up resilient networks for mirroring data, and achieving more efficient connectivity for development during natural disasters. Moreover, it helps identification applications to store archival data, speed up the performance, and unlock decentralized archiving. DHT can provide resilient access to data with low latency and dependence on connectivity and provide immutable records in transactions and time stamping. Blockchain is considered an extra layer combined with DHT to keep the immutable history of records, specifically for a transactional environment [5]. Furthermore, blockchains can be an identity certification authority where a smart contract can improve identification. The blockchain can store public keys Decentralized Identifiers (DIDs), where this model appears specifically in uPort and ShoCard. This scheme works similarly to a phonebook that anyone can use to verify specific public entities. When someone wants to verify the authenticity or validity of the credential, they can check the stored DIDs on the blockchain to see who issued it without contacting the issuer. Therefore, the blockchain is considered to function as a verifiable data registry.

This article's main contribution is measuring and comparing the decentralized identification solutions' performances. Furthermore, this research presents different decentralized identification systems by incorporating DHT as a solution to deploy decentralized applications with different aspects, such as biometric identification (machine learning) and other SSI technologies, such as uPort and ShoCard [6]. The measurement and comparison of the performance of different identification systems, such as serverless, server-, cloud-, SSI-, blockchain- and DHT-based systems, are discussed in this article. We consider throughput, execution time, and average standard deviation as three main parameters to calculate and measure the performance in four different models.

The rest of this article is organized as follows. In Section II, we introduce our scheme's related work. In Sections III and IV, we present the background and definitions. Section V describes different models. Section VI provides the specific comparison and measurements. Finally, we discuss and conclude the paper in Sections VII and VIII.

II. RELATED WORK

Identification systems have changed impressively in recent years with many upgrades. Although users typically should be approved by identifying themselves before joining, Golosova and Romanovs showed how to prevent unknown guests from joining [7].

Biometrics recognition is an intelligent solution that uses machine learning algorithms for identification and authentication, such as face recognition, fingerprinting, and IRIS, which are in the centralized identification category [8]. Security and immutability, as two properties of blockchain and DHT, can be used as backend technologies for identification systems. Alizadeh *et al.* [5] found that new decentralized applications combining blockchain, DHT, and ML can improve older systems in order to achieve better performance.

Fersi *et al.* [9] showed that DHT-based systems could increase the system performance by adding a fast lookup in large-scale deployment among decentralized systems. Alizadeh *et al.* [2] explained that a blockchain-based system could increase system security, specifically Internet of Things type of applications.

Navas and Beltrán [10] explained how different federated identification techniques use third parties with their identification solution. Cloud services with distributed centralized architectures, such as Facebook and Google, are well-known examples. Self-sovereign identity is a new generation identification solution. Belchior *et al.* [11] presented digital identification systems that deliver the strength to users to manage their identity data, the credibility of disclosed identity data, and network-level anonymity. Users' privacy is one of these systems' properties. Kim *et al.* [12] systemically explored key components of DID systems and analyzed their possible vulnerabilities when deployed. Liu *et al.* [13] showed different self-sovereign open-source identity management systems provided to users, organizations, and other entities. For example, ShoCard is a self-sovereign digital identity system that protects consumer privacy. Additionally, the authors explained how an identity platform could be built on the blockchain by showing a driver's license and how it can be so secure that a bank can rely on it. Shuaib *et al.* [14] analyze and evaluate the existing SSI solutions and develop the best possible solution for a blockchain-based land registry system. Furthermore, the authors investigate each SSI solution and present its advantages and limitations. Alzahrani [15] combined the decentralized features and the ''lookup by name'' property with a secure mechanism for maintaining synchronized replicas of an item in multiple locations to achieve short lookup times.

III. TERMINOLOGIES AND DEFINITIONS

This section defines different terminologies related to identification technologies.

- A **user or entity** on the Internet is a person, organization, computer application, thing, or smart device digitally connected to a network. An entity recognized by a unique property can be authenticated and eventually authorized in case of requesting access to online resources. Therefore, each entity has its digital identity.
- An **attribute** is a characteristic of an entity. For example, attributes might be permanent (such as a person's birth date), temporary (such as an address), or long-term (e.g., social security number).

- A **digital identifier** is a collection of information used to represent, analyze, and authenticate an entity in a digital environment without the intervention of human controllers after identifiers finish their duty [16]. Identifiers are characterized as a set of features that are referred to as identifiers to identify the entities. An identifier usually takes the form of a name or an address. An identity management system (IMS) is also used for enterprise or cross-network identity management.
- **Identification** is known as a process of the pairing of IDs with an entity presenting qualities [17]. Examples include associating a physical person with a claimed name, attaching a firm to a financial record, or relating a patient to physical characteristics.
- **Authentication** is known as the process of supplying sufficient credentials to confirm an entity's identification [17]. When a user enters the correct user name and password, for example, he or she establishes account ownership. Authentication mechanisms include PIN codes or passwords, identity cards, credit cards, smart cards, security tokens, mobile phones, ID documents, fingerprints, faces, irises, motor skills, gestures, and keystrokes.
- **Authorization** is the process of identification when authentication is completed successfully [17]. An entity's authority can be given based on its established identification. As a result, certain activities can be permitted depending on entity properties. Examples include a person's capacity to claim credit lines or an emergency vehicle's permission to cross past a red signal.

IV. TECHNOLOGIES AND BACKGROUND

This section briefly explains the current identification systems. Additionally, it describes different technologies, such as blockchain, DHT, uPort, and ShoCard.

A. CENTRALIZED, FEDERATED, AND DECENTRALIZED IDENTIFICATION

Many platforms use different identification architectures, which are divided into centralized [18], federated [19], and decentralized [20]. The use of centralized identity systems is nowadays common, and the typical paradigm. Typically, individuals use the services of an organization which maintain or own the identity system. The system's owner acquires, keeps, and utilizes the individual's identification. Private organizations such as banks, social media corporations, and governments already maintain similar systems. A matching verification now makes most authentications use login username and password. A digital account is often generated by the user and kept in a service provider's database. Typically, a user has one account for each service provider.

However, creating several identities in many systems, such as social media sites, is simple. This technique has enabled a digital representation of an entity, allowing for a wide range of online services. However, because of third-party data control, individuals' privacy might be at risk, and their online activity could be connected and eventually tracked. Furthermore, an exceedingly fragmented landscape will emerge because the user will be required to create a separate identity for each service provider. Finally, from the service provider's standpoint, such an approach requires a significant investment of resources to store, preserve, and safeguard users' data.

A federated identity system establishes mutual trust between centralized systems. A federated identity is accomplished by distributing verification and trust components across all identification systems or by mutually accepting the standards used by each system. For example, international organizations or governments could agree to recognize each other's credentials. It is also possible for businesses to agree to accept each other's identity verification system. The owners of identification systems frequently use legal agreements and shared technological standards to build one-to-one trust. As a result, the network and its reputation rise as the number of trustworthy relationships grows.

Users frequently prefer the simplicity of federated identification while accessing numerous services on different platforms, resulting in the widespread use of federated systems. On the other hand, building trust between two or more system owners is not always simple. The same applies to centralized systems, where the degree of trust depends on the system owners, the identity verification degree, and the data vetting process. Many web services propose identifying with Google or Facebook accounts to use their services such that these providers perform the user's identity verification.

Moreover, multi-factor authentication [21] is widely used as an extra security layer to make systems recognize that the people trying to gain access to an online account are who they claim they are. E-identification is another example that is typically used by some governments which are limited to a certain country or geographical region. Different e-services usually require that persons have Swedish electronic identification. E-identification is equivalent to other standard forms of ID, such as a driving license and a national identity card. Moreover, it allows people to identify themselves or sign a document or transaction securely online [22]. These responsibilities, risk allocation, and the formation of technical standards add complexity for system owners. In addition, these issues may result in high implementation costs, which typically lead to the lack of a variety of services consumers desire [23].

In summary, centralized and federated identification is referred to as classical systems since identity attributes are managed by a third party, such as an identity provider.

Decentralized identification is a technology that is handled with the help of all participants. It has a different architecture compared to centralized and federated identification services. There is no single organization inside to manage identification [24]. Usually, a decentralized identifier works in a peer-to-peer network, such as DLT and DHT. Decentralized identification systems are formed by many nodes that can be users, organizations, issuers, and validators. Self-sovereign identity systems represent a kind of this system. They operate

to manage digital identities where the users themselves manage attributes.

B. SELF-SOVEREIGN IDENTITY

Self-Sovereign Identity (SSI) provides people with authority over their digital identities in a decentralized manner [25]. SSI refers to a new identity management system in which the user retains complete control over his or her identity data without the need for outside interference. Figure 1 shows different entities in the SSI cycle. For example, in a transaction, one party will submit credentials to the other parties, and the other parties will verify that the credentials originated from a trusted issuer. The verifier's confidence in the issuer is passed to the credential holder.

Anyone can show verifiable credentials, and the person or entity confirming the credential decides whether to trust the entity that issued it. It is similar to a store clerk determining whether to accept a driver's license as evidence of age when buying alcohol.

Users have ownership over their verified credentials, and their permission is necessary to utilize them. This permission minimizes the unintentional disclosure of users' personal information. This feature is contrasted with the centralized identity paradigm, where some third parties provide identity.

Holders produce and manage unique IDs known as decentralized identifiers in an SSI system. For example, data from an issuer's database, a social media account, a history of purchases on an e-commerce site, or testimony from friends or coworkers might all be included in the credentials.

The "trust triangle" describes the basic structure of SSI with three participants. A person, an organization, and a smart device can play any role in the triangle [26].

Credential holder-Issuer-Verifier (Trust Triangle): A credential holder is an entity that has a license, permission, certificate, or registration issued by the government or a board being referred to as a credential holder. Additionally, a person who has a pending application for a credential for not more than one year from the date the application was filed to the department is referred to as a "credential holder". An entity can play a role by having one or more verified credentials and using them to create presentations. Credential repositories represent a place where holders save their credentials. The issuer is the entity that creates the credential. The verifier is an entity's role when receiving one or more verifiable credentials for processing, which may or may not be contained within a verifiable presentation. A verifier verifies the integrity of the supplied verifiable presentation and verifiable credentials. This process should involve checking the status of the verified credentials for revocation.

Validation means proving that a verifiable certification or verifiable presentation fulfills a verifier's and other stakeholders' requirements. Accordingly, the scope of this specification does not include validating verified credentials or verifiable presentations.

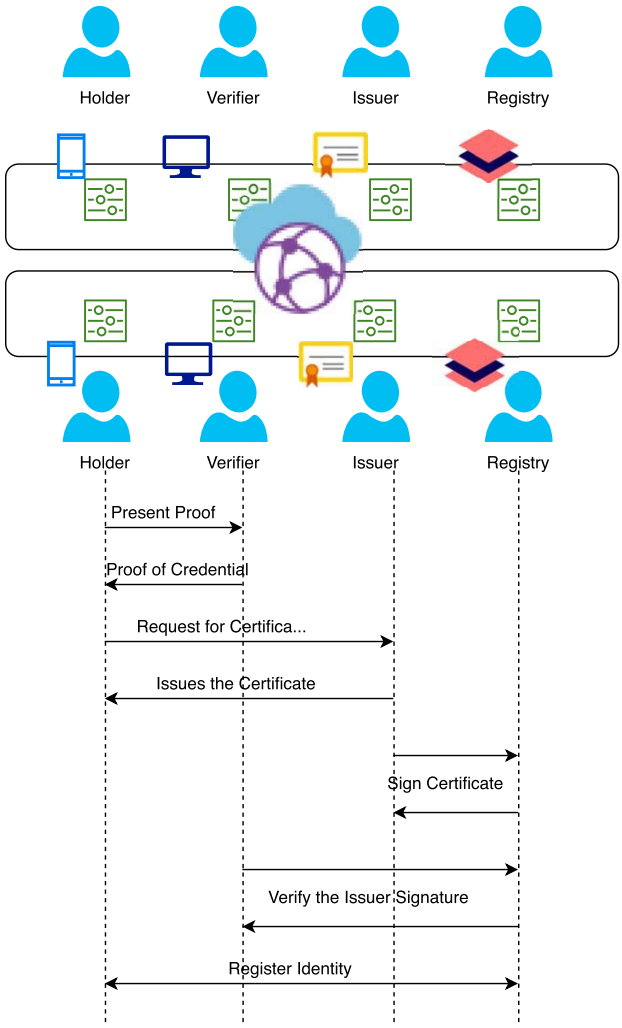


FIGURE 1. SSI schema: An issuer issues a certificate to a holder. The holder presents it to a verifier. The verifier verifies the requests. The registry records the event details.

Decentralized Identifiers (DIDs) are identifiers for decentralized systems where users can have verified digital identities [3]. They are introduced to the concept of self-sovereign identity. A DID identifies any entity. These identifiers allow a DID controller to demonstrate control over it. They may be used without a centralized registry, identity provider, or certificate authority. DIDs are Uniform Resource Identifiers (URIs) that link a DID subject to a DID document. An example of a DID is `did:example:123456abcdef`.

Decentralized Identifier Document is a document that is accessible through a verified data registry that contains information connected to a specific decentralized identification, such as the associated repository and public key information, and is also known as a DID document [3].

C. DISTRIBUTED HASH TABLE (DHT)

A distributed hash table (DHT) is a decentralized data store component that is a fast method among other decentralized

systems because of the rapid lookups of data based on key-value pairs. For example, the Interplanetary File System (IPFS) is a platform designed based on DHT [27]. It helps identification applications to store archival data, slash bandwidth costs by secure and peer-to-peer content delivery, speed up the performance, and unlock decentralized archiving.

DHT is a decentralized technology with a fast lookup time compared to other decentralized systems. The user can obtain the appropriate file according to the unique hash address. IPFS enables the storage of decentralized data without the need for additional memory. Furthermore, IPFS is a decentralized file-sharing network. IPFS includes both command line and graphical user interfaces, making it simple. It stores data packages by supplying data chunks. The distributed component of DHT, on the other hand, indicates that the complete table is shared over many places. IPFS recognizes which nodes have which data by using Kademlia technology. Kademlia was created in 2002 by Maymounkov and Mazieres and is based on DHT for decentralized peer-to-peer computer networks [4]. IPFS is capable of storing data regardless of its size. The hash address of the captured data may then be used to retrieve the data. The data will be broken into several small pieces, each of which will be recognized by its hash address. These chunks are dispersed to nodes with hashes nearest to the node. When the system receives a query request, all chunks may be concatenated to reconstruct the main object after viewing all minor bits.

Furthermore, IPFS can provide resilient access to data independent of latency and connectivity and be immutable by providing permanent links in transactions and time stamping [28]. Securing content without on-chain storage of the actual data is the main feature of IPFS. Blockchain specialists can use the content addressing features of IPFS for off-chain storage of large files and replace immutable, permanent links in transactions.

D. BLOCKCHAIN

Blockchain is a distributed ledger technology. Blockchain as a decentralized system provides an immutable decentralized processing environment for applications. A blockchain arranges recorded data into blocks chained together, although distributed ledgers record, share, and synchronize transactions in their digital ledgers using independent computers. One of the critical aspects of blockchain is storing a cryptographic signature of recorded data and events [28]. Blockchains also help to protect transactions' data from being changed. In decentralized systems, permissioned and permissionless blockchains are two forms of blockchain technology. In permissioned blockchain systems there is a limited number of known trusted participants carrying a copy of the blockchain's ledger.

Permissionless blockchain refers to a system that allows anyone to join or cancel their account. A well-known example of a permissionless blockchain is Bitcoin. It manages decentralized digital money without relying on a central authority. The blockchain is made up of blocks and data packages that represent the historical data of transactions. The main blockchain property is that it has unique timestamps and hash values. Each block is linked to the previous block, referred to as a parent block. It is possible to return to the first or genesis block by following the parents. Most network participants approve a new block using their consensus process, which is added to the validated block list. The information will be disseminated to several or all connected parties. After the consensus procedure is completed, all nodes that received the data will replicate and save an exact copy of the transaction information. This information is maintained individually on each node, resulting in trust between them. The use of crypto or credits to pay for activities and transactions is required. These credits incentivize participants to reach an agreement, also known as proof of stake or labor and receive money from the transaction's commission. Participants in the network are also encouraged to compete to win extra credits. Ethereum is a permissionless blockchain that uses smart contracts to operate, where Ether is its currency [29]. A smart contract is a Solidity-based computer program or transaction mechanism. It can carry out legally relevant events and activities automatically following the provisions of a contract or agreement. Figure 2 shows the interaction among users in the smart contract schema. First, the owner publishes an accommodation with the rental fee on the ledger. Then, other users, such as Renter, can see the different announcements on the web portal. Next, the smart contract will be executed when a Renter accepts the contract terms. The transaction includes information such as Owner, Renter, Signatures, and two side addresses. The timestamp will be stored and shared through the network participant as a copy of the ledger. Then, all network members keep a copy of the proof and know who rented it, when it was rented and to whom the accommodation was rented. This asset will never be removed and changed during the network lifecycle.

E. uPort

uPort is a way of registering identity by the help of the Ethereum blockchain. It enables users to identify themselves and send information to others in a clear, transparent way [30]. Figure 3 shows three scenarios with uPort. Figure 3 shows three scenarios with uPort. The first scenario is about registering a new user to the network with the help of the mobile application. Then, all participants (users, developers) who want to be involved must be registered with the application. The second scenario shows how a developer registers an application to the network. Developers need to be registered before registering an application to the network. Then, the developer who creates decentralized applications (DApps) [31] must be approved by the uPort application (network). Finally, the users' communication will start after the uPort's identification is finished in the third scenario. They need to identify themselves to log in to the DApps. They will start after the developers verify the Dapps and fix the connection links provided by the uPort developer portal. uPort uses QR code technology to provide a better

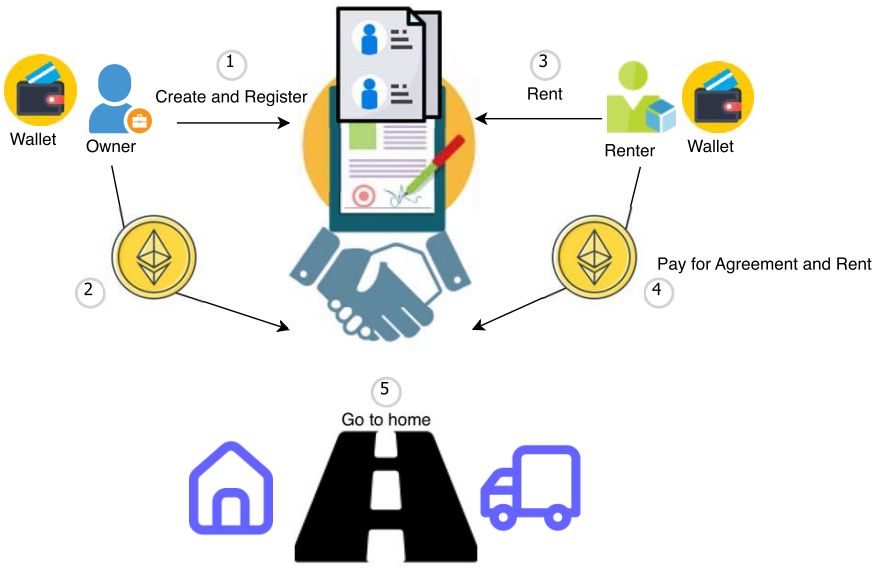


FIGURE 2. Demonstration of how the Ethereum smart contract can be executed.

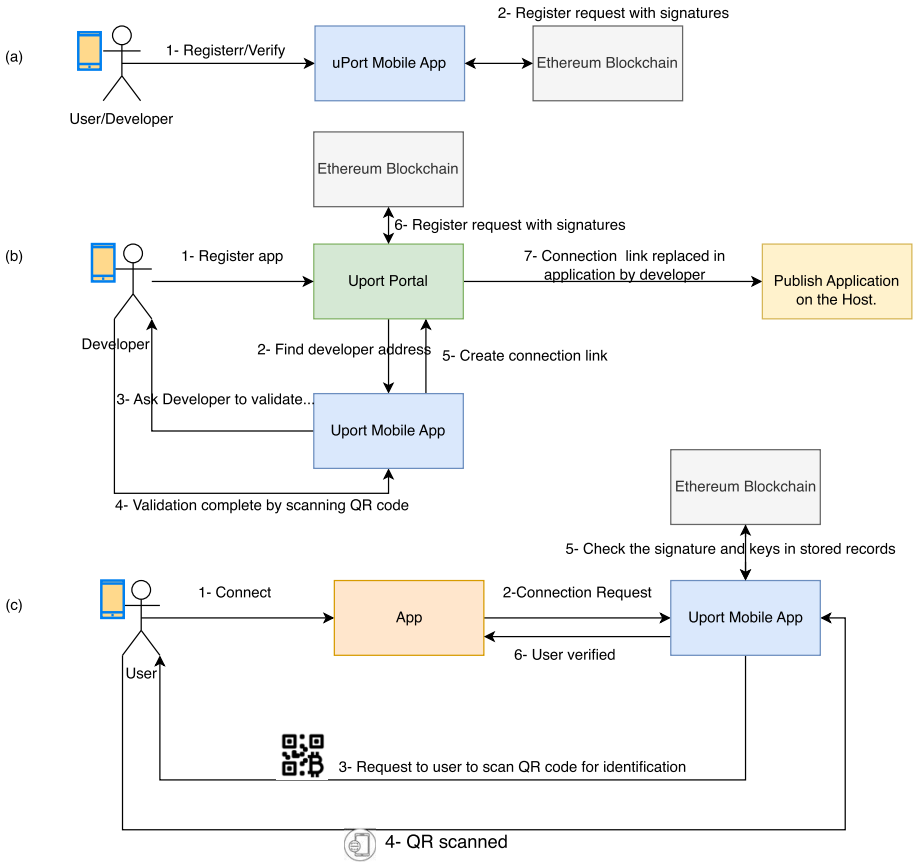


FIGURE 3. uPort schema (a) Register a user. (b) Connecting and identifying a mobile app. (c) Identifying users to enter the system.

and easier phone link to the system for both users and developers.

The idea behind uPort is that the blockchain can bypass the critical problem of keeping private keys. Instead, uPort uses

the blockchain as an identity certification authority where a smart contract represents the digital identity while allowing the revocation and replacement of that user's keys. The blockchain's participants can verify the data's originality and track the data source records. Furthermore, the blockchain addresses the effective traceability and access control of data.

F. ShoCard

ShoCard is another SSI technology which is decentralized. Users are the owners of their identities, regardless of the ownership of login to the applications. Without their permission, no one should view or use their data [32]. The ShoCard wallet gives individuals complete control over their data, allowing them to share just the information they want with others while keeping their personal information safe on their phones. Users can install the wallet and take complete control over their data by sharing only certificates. The current version asks users to register their driving license as an image with others while securely storing their personal information on their mobile devices. It is similar to uPort and has three phases: 1) certifying users; 2) adding connection links to DApps; and 3) user interaction with their wallet installed by their phone. In addition, it supports QR code technology to better match the phone with the systems. The ShoCard platform secures a person's biometric information and government ID to the private key on the device to sign a document, which binds it to the digital ID stored on the blockchain.

G. CLOUD SERVICE AS A HOST

A cloud represents a distributed centralized system managed by an organization and provides a pool that performs tasks with the help of installed services. Those resources hosted and delivered over a network are typically Internet-based and accessed on demand by multiple users. In addition, the cloud can provide servers to manage the functions for delivering processing power, storage, and applications. Then, it is possible to upload scripts and applications to the server as a host for deployment.

H. IPFS AS A HOST

IPFS [27] is a peer-to-peer hypermedia protocol that replaces old HTTP and makes the web faster, safer, and more open. It is decentralized and uses a similar peer-to-peer protocol to BitTorrent and a versioning system similar to Git. As a result, developers can build a static website, and IPFS deploys the uploaded application on DHT for free on decentralized hosting.

I. PRIVACY AND TIME MEASUREMENT

People increasingly conduct business, socialize, and communicate through the Internet. They need to protect their privacy and control what information they share online. Going forward, a move to decentralized identity solutions is essential to ensure data privacy and security. Private information can be shared securely by using decentralized solutions such as blockchain. Users can remain under complete control of their data. In contrast to centralized solutions, decentralized systems ensure that private data remain immutable and secure and can only be shared when selected users consent to provide information. Figure 4 shows a relation between response time and privacy in three different architectures for hosting identification: 1) Internet service providers as centralized host; 2) the cloud as a distributed host; and 3) DHT as a decentralized host. The throughput typically increase by changing from centralized to decentralized. One solution is using IPFS as a server to compile identification applications. It supports static and serverless identification versions to compile.

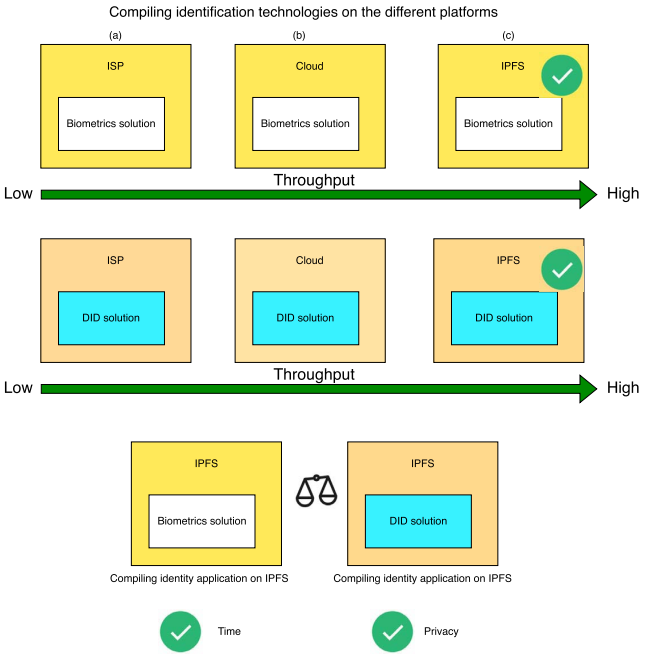


FIGURE 4. Relation among three different architectures: (a) Architecture based on ISP (b) Architecture based on Cloud services (c) Architecture based on DHT (IPFS).

J. PERFORMANCE

A performance study is the systematic explanation of the action or process of performing a task or function. The performance can depend on different parameters. For example, application performance is calculated based on user accessibility and response time. This article defines high performance as the models' output providing high accessibility, immutability, throughput, and trust. Most of these parameters are achievements of combining blockchain and DHT in a decentralized manner. The rate of successful message delivery through a communication channel such as Ethernet or the Internet is called network throughput. Less execution time and high throughput are two independent properties of high-performance models. A system with these properties also can normally manage larger network sizes and tasks (scalability).

V. MODELS

In this section, we define and illustrate different identification models based on different technologies. A classical model

is an easy form of identification. The architecture is of the simple client-server type, as ISPs provide this type of service. In this model, biometrics is used for identifying a user. The identification process consists of detecting biometrics such as face detection and face recognition by finding the match using the stored identities in the database. Also, the centralized server has the power in hand. The performance in these systems is defined by server management, which depends on server-side performance and throughput.

A. IPFS AS A HOST (BioIPFS)

This model is a decentralized form of identification. The architecture is DHT-based, where IPFS provides this type of service. In this model, biometrics is used to identify users. The identification process consists of detecting biometrics such as face detection and face recognition by finding the match by the stored identities detailed in IPFS-based data storage. The performance in these systems is defined by the number of nodes, the Internet bandwidth, and which gateways are used. Furthermore, no centralized server in the middle has the power in hand. The process starts with face detection hosted on IPFS. This module uses TensorFlow to detect a human face in an image. In the second step, it should find a matching face image description in the data storage, such as Orbitdb, as an IPFS-based database. Then, the user can communicate with the system hosted on the IPFS after finding the user description in the data storage.

B. BLOCKCHAIN-BASED IDENTIFICATION BIOMETRICS HOSTED ON IPFS (BBID)

In this model, IPFS helps to load the identification application. The combined machine learning, face recognition solution based on a public blockchain, and IPFS can be a good idea for identification. The event's information, such as addresses and timestamps, will be stored as a transaction in this model, which manages transactions using its immutable historical records. This process makes it secure and trusted for all users. Moreover, supporting decentralized web hosting and sharing data in the IPFS can be beneficial for identification among decentralized systems. IPFS' role is to compile the identification software as a web application and allocate resources such as memory and storage to a virtual server. The main difference between this and the BioIPFS model is that blockchain is a record-keeping extension through its immutable transactional environment.

C. uPort HOSTED ON DHT

In this model, IPFS loads the uPort-based application. Then, IPFS compiles the identification software as a web application and allocates resources such as memory and storage to a virtual server. Here, users who have already installed the uPort user application on their smartphone try to log in to the application hosted on the IPFS. In addition, developers must register the application to the uPort portal as an initial step and fix the scripts in their application. These specific scripts and the QR algorithm are provided by the uPort portal based on the developer registration only for that application. Therefore, all applications have separate and different codes. In the second step, the user must scan the QR code provided by the developer on the web application and send the agreement from the phone to the application through the uPort mobile application after receiving a notification on their phone that asks permission to access some part of the information. The main differences between this model and BioIPFS and BBID are that uPort is replaced with biometric recognition, and the blockchain does not need to act as a record-keeping module.

D. ShoCard

In this model, the IPFS service helps to load the ShoCard application. IPFS' duty is to compile the ShoCard software as an application and allocate resources such as memory and storage to a virtual server. Here, developers must fix the connection scripts in their applications. The ShoCard website provides these scripts and QR algorithms to the developers. Then, users should have the ShoCard application installed on their smartphone and register to the mobile application with their face and driving license images. Then, after registration, users should be logged into the application hosted on IPFS. In the next step, the user must scan the QR code provided by the developer on the application and approve the connection by their phone through the ShoCard mobile application after receiving a notification on their phone that requests allocation to access user information. The main difference between this and the previous model is that ShoCard is replaced with a uPort with a different structure background, biometric recognition, and no immediate need for the blockchain as a record-keeping module.

VI. EVALUATION

In this article, we divided the evaluations into two parts. The first part's main target is to compare three different hosts for a decentralized application and measure the performance between cloud-based and IPFS-based versions. Microsoft Azure VM was used as a cloud-based service and `cf-ipfs` as a gateway for IPFS. A Linux/amd64 Ubuntu server 20.04 LTS Gen2 was installed on Azure VM with the Docker v20.10.7 deploy Node.js application on port 80, where we set up AZURE Service Standard-B1s with 1 CPU holding 1 GB memory and Azure service Standard-D4s-v3 with 4 CPUs and 16 GB memory. Apache JMeter 5.4.1 was the application used to measure performance.

Tables 1 and 2 display different tests to show the hosts' performance in the cloud and IPFS. Table 1 shows load testing with 100 samples, while Table 2 shows load testing with 1,000 samples. The results present the hosts' strengths by deploying the same application on three hosts. We choose the fastest gateway between IPFS gateways (`cf-IPFS.com`) to compare with the MS Azure cloud service. Additionally, we choose central Sweden as the main resource backend and storage for MS Azure and the closest MS Azure datacenter.

In the second part, we compare the execution time of the four discussed models. JavaScript (npm 6.14.16 and

TABLE 1. Load testing with 100 samples (request), Ramp up: 10, loop: 1.

Hosting Services	Error %	Throughput Requests/Sec	STD Deviation	Received KB/Sec	Sent KB/Sec	Avg. Bytes
Azure Service Standard-B1s	0.0%	9.7	4.82	1.21	1.08	128
Azure Service Standard-D4s-v3	0.0%	10.1	3.06	1.26	1.13	128
IPFS (cf-ipfs.com)	0.0%	10.1	10.80	65.69	3.37	6692

TABLE 2. Load testing with 1000 samples (request), Ramp up: 10, loop: 1.

Hosting Services	Error %	Throughput Requests/Sec	STD Deviation	Receive KB/Sec	Sent KB/Sec	Avg. Bytes
Azure-Standard-D4s-v3-Docker	0.0%	97.6	5.30	196.55	11.06	2062
Azure Service Standard-D4s-v3	0.0%	97.7	5.47	195.75	11.07	2051
IPFS (cf-ipfs.com)	0.0%	98.6	32.89	644.72	33.03	6695

TABLE 3. Performance comparison among models - Response Time (ms) (Model V-A) DHT-based (Model V-B) DHT- and Blockchain-based (Model V-C) uPort-DHT (Model V-D) ShoCard-DHT (Model V-A σ^2): DHT-based Variance (Model V-B σ^2): DHT- and Blockchain-based Variance (Model V-C σ^2): uPort-DHT Variance (Model V-D σ^2): ShoCard-DHT Variance.

10 repetitions	Model V-A	Model V-B	Model V-C	Model V-D	Model V-A σ^2	Model V-B σ^2	Model V-C σ^2	Model V-D σ^2
	4456	6544	7039	10238	637,9	2417,3	1908,2	811,1
	3792	11480	6187	12638	665,2	2464,6	2006,7	676,1
	4503	7987	6206	11972	653,4	2243,3	1955,7	623,1
	3804	7840	6256	11529	682,9	2377,4	2006,3	657,4
	3443	8617	7482	11899	716,7	2535,3	2053,9	696
	3564	10720	6743	11291	698,9	2735,8	2206,5	750,6
	4008	9308	8046	12382	669,3	2751,1	2325,1	785,5
	5783	2473	7588	10883	701,3	2959,4	2593,6	833,5
	4397	10299	5948	11332	158,1	860,5	2940,5	762,7
	4011	8578	12793	13111	82,5	96,7	2682,1	691,75
Average μ	**4176**	**8384**	**7429**	**11728**	**567**	**21447**	**2268**	**729**

node.js 14.19.0) was selected as the programming language. uPort and ShoCard mobile applications were used to communicate with the uPort 1.7.6 and ShoCard 1.0.4 developer portals. Ethereum-Kovantest is a public test blockchain network for a developer and was used as a base for the model described in section V-B, where the Remix IDE editor was used for compiling and running the smart contracts written in the Solidity language. A Metamask extension was used as a wallet that includes accounts, addresses, and credit (ETH). Finally, Kovantest-Fucent was used to provide credit to communicate via the public blockchain. Ten experiments were executed, and the results are shown in Table 3 and Figure 5, where each model's average and variance for the functions mentioned earlier are indicated.

We used a MacBook Pro (15-inch, 2018) laptop with the following configuration: macOS Big Sur Version 11.6, 2.2 GHz 6-Core Intel Core i7 processor, 16 GB 2400 MHz DDR4 memory, 95/91 Mbps average download/upload data rates, Google Chrome Version 96.0.4664.110 (Official Build) (x86_64), IPFS Version 0.18.1, go-ipfs 0.11.0 `face-api.js` Version 0.17.0, and react-jitsi Version 1.0.4.

FIGURE 5. Different execution time for identifying users to enter the system.

VII. RESULTS AND DISCUSSION

We consider 100 and 1,000 as the number of virtual users per request in these tests. Therefore, the ramp-up value is ten, which means that JMeter will take 10 seconds for all 100 and 1,000 threads to be up and running. We tested two different configurations of cloud VM services with IPFS. Additionally, we compared different parameters, such as error, which shows the percentage of failed requests, and standard deviation is the set of exceptional cases that deviated from the average value of the sample response time. Finally, throughput is the number of requests that were

TABLE 4. Nomenclature table.

Terms	Meaning
μ	Average
σ^2	Variance (measure of variability)
Avg.Bytes	Average of total bytes of data downloaded from server
AWS	Amazon Web Services
BBID	Blockchain, Biometrics Hybrid Solution
BioIPFS	Biometrics Solution Hybrid IPFS
DApps	Decentralized Applications
DHT	Decentralized Hash Table
DID	Decentralized Identifiers
DLT	Distributed Ledger Technology
Error%	Denotes the percent of requests with errors
ETH	Ethereum (cryptocurrency)
IPFS	InterPlanetary File System
ISP	Internet Service Provider
ML	Machine Learning
QR Code	Quick Response code
Ramp up	Period to the full number of threads chosen
SSI	Self-Soverign Identity
STD.Dev	Standard Deviation
Throughput	(number of requests) / (total time)
URI	Uniform Resource Identifiers
VM	Virtual Machine

processed per time unit (seconds, minutes, hours) by the server, which was calculated from the start of the first sample to the end of the last sample, and Send/Receive rate is the amount of data uploaded/downloaded from the server during the performance test execution. For the case of 100 samples, DHT-based and cloud-based systems have almost the same performance, as shown in Table 1. However, for the case of 1,000 samples, the results are shown in Table 2, where the DHT-based solution has better throughput than the cloud-based system being dockerized and not dockerized. Therefore, DHT has shown better throughput during our tests. Table 4 shows terms and definitions for the tests.

Table 3 and Figure 5 show the different models mentioned earlier by repeating them ten times. The average of ten repetitions shows that Model V-A was executed in a shorter time than the other three models. Model V-D has the longest time needed to be completed because the blockchain module is a time-consuming part in Model V-D. Models V-A and V-B have lower variances in their execution times close to the average execution time. The low variance means that all repetitions were completed with less fluctuation in execution time consumed, representing more stable systems in executing and timing.

Model V-A is fast because of the DHT lookup function. However, Models V-B and V-C are slower because of forwarding requests to multiple modules, such as the holder, issuer, and validator. Model V-D is slower because of many references to different modules, such as certifiers and validators.

The overall benefit of our evaluation is to compare some existing solutions in principal and to perform some quantitative comparative performance evaluation. However, the performance evaluation is limited to small scale and full scalability evaluations is left as future work.

VIII. CONCLUSION

Identification is essential for providing user identity to applications that fulfill the standard requirements of decentralized systems. As a result, identification should meet minimum requirements to capture users' trust.

This article started with a definition of decentralization and decentralized identification. We also highlighted how the DHT's immutability and speed could be a helpful mechanism for managing identification. Then, we addressed how to provide decentralized application hosting and data sharing in a DHT-based architecture. The IPFS-decentralized web hosting solution is also an excellent way to keep all systems decentralized while avoiding memory constraints.

This part compared different types of identification in a decentralized environment. As one of the new approaches, we consider SSI technology and a machine learning facial recognition system-based solution, tuned with public blockchain and DHT technologies for better performance. Additionally, we showed that a system deployed on IPFS has better throughput than a system deployed on cloud services.

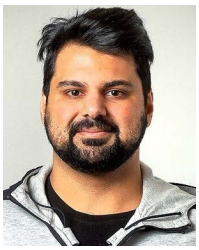
Although the combination of DHT and SSI effectively assists identification in a decentralized manner, they consume a lot of energy and time. Blockchain is an excellent solution for keeping the record of identities immutable but still has a significant problem. It is slow when it calls and decodes a query of many transactions' data in a large-scale system.

We plan to introduce additional decentralized techniques for strong identification with tuning the system by adding smart devices as entities to be identified by the system. We are also considering expanding our research to include device identification and human reaction and response time measurements when using these devices as handheld technologies.

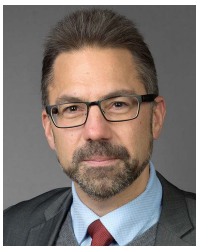
REFERENCES

[1] P. Kaushik, A. M. Rao, D. P. Singh, S. Vashisht, and S. Gupta, "Cloud computing and comparison based on service and performance between Amazon AWS, Microsoft azure, and Google cloud," in *Proc. Int. Conf. Technol. Advancements Innov. (ICTAI)*, Nov. 2021, pp. 268–273.
[2] M. Alizadeh, K. Andersson, and O. Schelén, "A survey of secure Internet of Things in relation to blockchain," *J. Internet Services Inf. Secur.*, vol. 10, no. 3, pp. 47–75, 2020.
[3] World Wide Web Consortium (W3C). *Decentralized Identifiers (DIDs) V1.0, Core Architecture, Data Model, and Representations*. Accessed: Jul. 22, 2022. [Online]. Available: https://www.w3.org/TR/did-core/
[4] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proc. Int. Workshop Peer–Peer Syst.* Springer, 2002, pp. 53–65.
[5] M. Alizadeh, K. Andersson, and O. Schelén, "DHT- and blockchain-based smart identification for video conferencing," *Blockchain: Res. Appl.*, vol. 3, no. 2, 2022, Art. no. 100066.
[6] P. Dunphy and F. A. P. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE Security Privacy*, vol. 16, no. 4, pp. 20–29, Jul./Aug. 2018.
[7] J. Golosova and A. Romanovs, "The advantages and disadvantages of the blockchain technology," in *Proc. IEEE 6th Workshop Adv. Inf., Electron. Electr. Eng. (AIEEE)*, Nov. 2018, pp. 1–6.
[8] A. Gelb and J. Clark, "Identification for development: The biometrics revolution," Center Global Develop., Washington, DC, USA, Work. Paper 315, 2013.
[9] G. Fersi, W. Louati, and M. B. Jemaa, "Distributed hash table-based routing and data management in wireless sensor networks: A survey," *Wireless Netw.*, vol. 19, no. 2, pp. 219–236, Feb. 2013.

[10] J. Navas and M. Beltrán, "Potential impacts in citizens' privacy of using federated identity management to offer e-government services," in *Proc. 16th Int. Joint Conf. e-Business Telecommun. (ICETE)*, 2019, pp. 350–355.
[11] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, and S. Guerreiro, "SSIBAC: Self-sovereign identity based access control," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1935–1943.
[12] B. G. Kim, Y.-S. Cho, S.-H. Kim, H. Kim, and S. S. Woo, "A security analysis of blockchain-based did services," *IEEE Access*, vol. 9, pp. 22894–22913, 2021.
[13] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K.-K. Raymond Choo, "Blockchain-based identity management systems: A review," *J. Netw. Comput. Appl.*, vol. 166, Sep. 2020, Art. no. 102731.
[14] M. Shuaib, N. H. Hassan, S. Usman, S. Alam, S. Bhatia, A. Mashat, A. Kumar, and M. Kumar, "Self-sovereign identity solution for blockchain-based land registry system: A comparison," *Mobile Inf. Syst.*, vol. 2022, Apr. 2022, Art. no. 8930472.
[15] B. Alzahrani, "An information-centric networking based registry for decentralized identifiers and verifiable credentials," *IEEE Access*, vol. 8, pp. 137198–137208, 2020.
[16] S. A. Williams, S. C. Fleming, K. O. Lundqvist, and P. N. Parslow, "Understanding your digital identity," *Learn. Exchange*, vol. 1, no. 1, pp. 1–6, 2010.
[17] D. Todorov, *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*. New York, NY, USA: Auerbach, 2007.
[18] J. Fang, C. Yan, and C. Yan, "Centralized identity authentication research based on management application platform," in *Proc. 1st Int. Conf. Inf. Sci. Eng.*, Dec. 2009, pp. 2292–2295.
[19] H. Yildiz, C. Ritter, L. T. Nguyen, B. Frech, M. M. Martinez, and A. Kupper, "Connecting self-sovereign identity with federated and user-centric identities via SAML integration," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Sep. 2021, pp. 1–7.
[20] L. Stockburger, G. Kokosioulis, A. Mukkamala, R. R. Mukkamala, and M. Avital, "Blockchain-enabled decentralized identity management: The case of self-sovereign identity in public transportation," *Blockchain: Res. Appl.*, vol. 2, no. 2, Jun. 2021, Art. no. 100014.
[21] I. Velásquez, A. Caro, and A. Rodríguez, "Authentication schemes and methods: A systematic literature review," *Inf. Softw. Technol.*, vol. 94, pp. 30–37, Feb. 2017.
[22] *Swedish E-identification*. Accessed: Jul. 22, 2022. [Online]. Available: https://bolagsverket.se/en/fee/e-services/swedish-e-identification-1.16393
[23] G. Bendiab, S. Shiaeles, S. Boucherkha, and B. Ghita, "FCMDT: A novel fuzzy cognitive maps dynamic trust model for cloud federated identity management," *Comput. Secur.*, vol. 86, pp. 270–290, Sep. 2019.
[24] A. Grüner, A. Mühle, T. Gayvoronskaya, and C. Meinel, "A comparative analysis of trust requirements in decentralized identity management," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Germany: Springer, 2019, pp. 200–213.
[25] A. Preukschat and D. Reed, *Self-Sovereign Identity*. Shelter Island, NY, USA: Manning Publications, 2021.
[26] G. Laatikainen, T. Kolehmainen, and P. Abrahamsson, "Self-sovereign identity ecosystems: Benefits and challenges," in *Proc. Scand. Conf. Inf. Syst.*, 2021, pp. 1–18.
[27] *IPFS Powers the Distributed Web*. Accessed: Jul. 22, 2022. [Online]. Available: https://ipfs.io/
[28] M. Alizadeh, K. Andersson, and O. Schelen, "Efficient decentralized data storage based on public blockchain and IPFS," in *Proc. IEEE Asia–Pacific Conf. Comput. Sci. Data Eng. (CSDE)*, Dec. 2020, pp. 1–8.
[29] C. Dannen, *Introducing Ethereum and Solidity*, vol. 1. Brooklyn, NY, USA: Springer, 2017.
[30] N. Naik and P. Jenkins, "UPort open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain," in *Proc. IEEE Int. Symp. Syst. Eng. (ISSE)*, Oct. 2020, pp. 1–7.
[31] W. Metcalfe, "Ethereum, smart contracts, DApps," in *Blockchain and Crypt Currency*, M. Yano, C. Dai, K. Masuda, and Y. Kishimoto, Eds. Singapore: Springer, 2020, ch. 5, pp. 77–93.
[32] A. Satybaldy, M. Nowostawski, and J. Ellingsen, "Self-sovereign identity systems," in *Proc. IFIP Int. Summer School Privacy Identity Manage.* Gjøvik, Norway: Springer, 2019, pp. 447–461.



MORTEZA ALIZADEH (Graduate Student Member, IEEE) received the master's degree in computer science (artificial intelligence) from Qazvin Islamic Azad University, Iran, in 2013. He is currently pursuing the Ph.D. degree in researching blockchain and smart identifications systems with the Pervasive and Mobile Computing Group, Luleå University of Technology, Luleå, Sweden. His research interests include blockchain, decentralized systems, smart contracts, and machine learning.



KARL ANDERSSON (Senior Member, IEEE) received the M.Sc. degree in computer science and technology from the Royal Institute of Technology, Stockholm, Sweden, and the Ph.D. degree in mobile systems from the Luleå University of Technology, Sweden. He was with the Internet Real-time Laboratory, Columbia University, New York, NY, USA, and the National Institute of Information and Communications Technology, Tokyo, Japan. He is currently a Full Professor of pervasive and mobile computing at the Luleå University of Technology.

OLOV SCHELÉN (Senior Member, IEEE) received the Ph.D. degree in computer networking from the Luleå University of Technology. He has more than 20 years of experience in industry and academia. He is currently a Full Professor at the Luleå University of Technology and the CEO of Xarepo AB. His research interests include mobile and distributed systems, software orchestration, computer networking, artificial intelligence, and blockchain.