ChainDiscipline

Towards a Blockchain-IoT-based Self-sovereign Identity Management Framework

Popa, Marius; Stoklossa, Sebastian Michael; Mazumdar, Somnath

Document Version
Accepted author manuscript

Published in:
IEEE Transactions on Services Computing

DOI:
10.1109/TSC.2023.3279871

Publication date:
2023

License
Unspecified

Citation for published version (APA):
Popa, M., Stoklossa, S. M., & Mazumdar, S. (2023). ChainDiscipline: Towards a Blockchain-IoT-based Self-sovereign Identity Management Framework. *IEEE Transactions on Services Computing*, *16*(5 (Sept.-Oct.)), 3238-3251. https://doi.org/10.1109/TSC.2023.3279871

ChainDiscipline - Towards A Blockchain-IoT-Based Self-Sovereign Identity Management Framework

Marius Popa, Sebastian Michael Stoklossa, Somnath Mazumdar
Department of Digitalization, Copenhagen Business School
Solbjerg Plads 3, 2000 Frederiksberg, Denmark
{mapo20ae, sest20ac}@student.cbs.dk, sma.digi@cbs.dk

***Abstract*—In today's complex Internet platform, online users need help to protect their online identity. Only sometimes, websites are very transparent about how user data will be collected, stored and processed by them. Sometimes Internet entities collect more online user information than required. These entities often share user identity-related data with third parties without consent. Existing traditional identity schemes need to be improved to stop and counter new ways of digital identity theft and fraud. Blockchain is a promising technology to strengthen the preservation of online users' digital identity due to its decentralised nature and robust data security features. In this paper, we proposed and implemented a generic blockchain-IoT-based self-sovereign identity management framework called ChainDiscipline. We have demonstrated the framework's operability and functionality by implementing healthcare and smart home data management-based use cases.**

***Index Terms*—Blockchain, Identity, IoT, Health, Smart Home**

I. Introduction

MOVING from Web 1.0 to Web 3.0, the Internet has become more complex, intelligent, and ubiquitous. Complex Internet infrastructure route user data traffic based on Internet service providers' (ISPs) business relationships [1]. It has been seen that not only the websites but also the ISPs collect user data for their financial gain. Data related to a specific user is generally stored on the service provider side in data silos. Users have no insights about how data is collected, stored, and processed, which makes the stored user data prone to exploitation without the user's consent. For instance, real-time user location data has also been shared with third parties for financial gains [2]. Web browsing and application usage data can be further exploited to link users' social media profiles for targeted advertisements [3]. A prominent example is political micro-targeting by a third-party (Cambridge Analytica) exploiting the personal information of 50 million Facebook users (without their proper consent). Furthermore, it is also possible to trace the host Ethernet link from the data packets [4]. Personal identifiable information is valuable for cyber-criminals, as it enables identity theft and fraud. In most cases, websites use cookies to collect data to track users further [5], while ISPs harvest user data for their enhanced/customised product or service offerings [2]. In most of these cases, the amount of data collected is too much and includes sensitive personal information.

Figure 1 shows the evolution of identity management concepts. It places importance on privacy as time progresses.

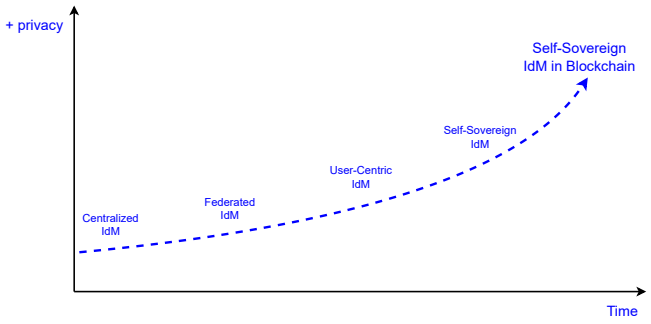

Figure 1. Evolution of identity management concepts [6].

As a result, the current digital identity concept incorporates multiple privacy-preserving features. After agreeing to obscure terms and conditions, identity holders have no guarantee of compliance by third parties. There is no way to control what trusted third parties do with user identity-related data. To take legal action, the first agreement violation has to be established. Moreover, financial compensation neither returns the user's identity nor prevents potential breaches in the future. It is the responsibility of a trusted third party to implement the required security measures. Users should also refrain from creating vulnerabilities with their casual approaches. Identity management systems (IdMs) can be essential in preserving privacy and improving security by controlling information flow among multiple entities, including third parties. Some countries made privacy-preserving rules or frameworks to protect the identity of their citizens. However, more has to be done in this regard. Notably, users should be informed and must retain control of their identity. Recently, the decentralized identifier has been established as a viable alternative to current digital identity concepts. In self-sovereign identity (SSI), users own their identity and can delegate it based on the requirement [7].

Blockchain has been proposed as a promising solution to counter privacy and security issues due to its robust security features (primarily digital signature and hashing) [8]. Blockchain can be seen as an unavoidable platform for driving self-sovereign identity and establishing new paradigms of digital identity [9]. Blockchain offers a distributed, cryptographically secure append-only shared ledger with a complete transactional history of user data. It can play an essential role as an accelerator for decentralizing digital identities. User identifiers stored in blockchain are less prone to identity

theft and unwanted data exploitation by third parties due to its strong data security features. Thanks to blockchain, third parties can quickly verify user identity without storing large amounts of user data.

Promising identity management applications focused on health data and smart homes have been proposed, leveraging the benefits of blockchain and Internet-of-Things (IoTs). Such IoT devices are connected to a local home network and smartphones via short-range wireless technology (such as Bluetooth and WiFi). These IoT devices differ in terms of device complexities, size, and application types. IoTs also come from different vendors with different firmware. Such dissimilarities bring challenges when updating IoT devices to fix bugs and improve security [1]. Thus, it can be seen that there are ways to lose sensitive personal data while using IoTs. However, properly implementing such devices and securely storing the data using blockchain can reduce the severity.

A significant part of the literature did propose and discuss potential blockchain-based self-sovereign identity management solutions for healthcare and smart home data management, but only a handful of those did provide details on their implementation (refer to Table I and Table II for comparison respectively). This paper aims to answer *How to design a generic blockchain-IoT-based self-sovereign identity management framework?* We aim to contribute to the existing literature by proposing and implementing a generic blockchain-IoT-based functional and robust identity management framework. We successfully mapped two use cases (healthcare and smart home) into this framework. In healthcare use case, collected data from IoTs can be processed to offer better patient-related services (such as faster diagnosis). Various stakeholders (such as hospitals, medical personnel, or health insurance companies) can benefit from the collected data. Meanwhile, the smart home aims to improve homeowners' comfort, convenience, and quality of life via implanting multiple smart sensors/IoT devices. The main contributions of this article are as follows:

- A Blockchain-IoT-based generic identity management framework has been proposed. We also explain the framework's core components (refer to Section III). The primary aim of this paper is to propose a framework and validate it.
- Both healthcare and smart home use cases are implemented using a four-node Hyperledger Fabric network with multiple IoT sensors to show the generability of the framework (mentioned in Sub-section IV-C).
- We performed a unit test to check the functionality of the important components (refer to Table IV).
- Finally, we also discuss the business aspect of the proposed framework (refer to Sub-subsection IV-E).

II. Related Work

The IdM domain has received a lot of attention from academia in past years [7]. We have performed a comparative literature study focusing on relevant IdM frameworks and prototype implementations based on fourteen evaluation criteria. Given the defined use case scenarios, the selected literature is from the application domains of healthcare and smart homes. Out of fourteen, we found that six criteria (identity holder consent, user control, provability, persistence, data protection, and transparency) are prevalent for both application domains. At the same time, key recovery features still need to be addressed.

A. Selection Criteria

The evaluation criteria cover multiple aspects of the four features: privacy, security, usability, and performance. For the privacy principle, the following criteria further have been defined: *i) identity holder consent*: the necessity of the users' consent for any access to identity-related data, *ii) user control*: the user can always access the data and determines who else can see and access it, *iii) provability*: the identity holder must be able always to prove his/her identity, *iv) pseudonymity*: the user does not have to disclose their real identity to interact within the IdM, and *v) persistence*: the users' identity exists until s/he initiates the removal. Concerning security, the criteria *vi) key recovery*: the keys and credentials can be recovered easily and safely, *vii) data protection*: all the information processed in the IdM is kept secure, and *viii) transparency*: the identity holder knows at any given moment the data access trail can be established, *ix) portability*: the ability to take one's digital identity anywhere (e.g., via the cloud or physical devices), *x) interoperability*: the bridging of identities to access web services and platforms, and *xi) IoT support*: the support for IoT devices in the IdM is the criteria linked to the usability principle. Lastly, performance-focused criteria are: *xii) scalability*: the viability of the IdM for further adoption and reproduction, *xiii) data minimization*: the minimal amount of data required is processed for any interaction in the IdM, and *xiv) transaction costs*: the financial costs for transactions on the blockchain. Regarding the criteria for performance, only the works which have been validated by simulation or by implementation are considered in this study.

B. Related Work Focuses on Healthcare

Table I listed the existing literature in the healthcare domain. From the table, it can be seen that none of the solutions completely fulfil all the fourteen criteria, but six important criteria (such as identity holder consent, user control, provability, persistence, data protection, and transparency) are fulfilled by all the selected works. Privacy and security are fundamental for healthcare-based IdM. It means the data (identity) owner should *i)* always be aware, and *ii)* in control of when and with whom the data is shared. Moreover, *iii)* the data owner can decide not to share the data anymore. In blockchain-based IdM, *vii)* the data is protected, and *viii)* all prior data accesses (or transactions) logs can be maintained.

Most studies focused on developing a working prototype rather than including advanced security features for blockchain-based applications. This might be an explanation as to why the criteria *vi) key recovery* has rarely been addressed in the literature. Roehrs et al. incorporated the lost access recovery mechanism into the authenticator component of the

[1]It is worth noting that most cheap IoT devices never get any update in their lifetime.

Table I
COMPARISON OF RELATED WORK IN HEALTHCARE

	Privacy					Security			Usability			Performance		
Work	**i**	**ii**	**iii**	**iv**	**v**	**vi**	**vii**	**viii**	**ix**	**x**	**xi**	**xii**	**xiii**	**xiv**
Ahram et al. [10]	X	X	X	-	X	-	X	X	P	P	X	X	-	-
Benchoufi et al. [11]	X	X	X	-	X	-	X	X	-	-	-	X	-	-
Bocek et al. [12]	X	X	X	-	X	-	X	X	P	P	X	X	-	-
Brogan et al. [13]	X	X	X	-	X	-	X	X	P	X	X	X	X	X
Chen et al. [14]	X	X	X	-	X	-	X	X	P	-	-	P	X	X
Dagher et al. [15]	X	X	X	-	X	-	X	X	-	X	-	X	-	-
Dwivedi et al. [16]	X	X	X	X	X	P	X	X	X	-	X	X	-	-
Griggs et al. [17]	X	X	X	X	X	-	X	X	-	-	X	-	-	-
Hang et al. [18]	X	X	X	-	X	-	X	X	-	X	X	P	X	X
Hasan et al. [19]	X	X	X	-	X	-	X	X	-	-	-	P	X	X
Ismail et al. [20]	X	X	X	-	X	-	X	X	P	-	-	X	X	X
Kaur et al. [21]	X	X	X	-	X	P	X	X	X	X	-	-	-	-
Li et al. [22]	X	X	X	X	X	P	X	X	-	X	-	X	X	X
Liang et al. [23]	X	X	X	P	X	-	X	X	X	X	-	X	X	X
Mikula et al. [24]	X	X	X	P	X	P	X	X	-	P	-	P	X	X
Nguyen et al. [25]	X	X	X	-	X	P	X	X	X	-	-	X	X	X
Rajput et al. [26]	X	X	X	-	X	P	X	X	-	-	-	X	X	X
Roehrs et al. [27]	X	X	X	-	X	X	X	X	-	X	-	X	-	-
Song et al. [28]	X	X	X	-	X	-	X	X	P	X	-	X	X	X
Thwin et al. [29]	X	X	X	-	X	-	X	X	X	X	-	X	X	X
Uddin et al. [30]	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Vora et al. [31]	X	X	X	-	X	-	X	X	-	X	-	-	-	-
Xia et al. [32]	X	X	X	-	X	P	X	X	X	-	-	X	X	-
Zhang et al. [33]	X	X	X	X	X	-	X	X	-	X	-	X	X	X
Zheng et al. [34]	X	X	X	P	X	-	X	X	X	X	X	X	X	X

1 = Identity Holder Consent, 2 = User Control 3 = Provability, 4 = Pseudonymity,
5 = Persistence, 6 = Key Recovery, 7 = Data Protection, 8 = Transparency, 9 = Portability,
10 = Interoperability, 11 = IoT Support, 12 = Scalability, 13 = Data Minimization,
14 = Transaction Costs

X = Criteria is completely fulfilled, P = Criteria is partially fulfilled,
- = Criteria not fulfilled / not applicable

Table II
COMPARISON OF RELATED WORK IN SMART HOME

	Privacy					Security			Usability			Performance		
Work	**i**	**ii**	**iii**	**iv**	**v**	**vi**	**vii**	**viii**	**ix**	**x**	**xi**	**xii**	**xiii**	**xiv**
Aggarwal et al. [35]	X	X	X	-	X	-	X	X	X	X	X	X	X	X
Arif et al. [36]	X	X	X	-	X	-	X	X	X	P	X	P	X	X
Bouras et al. [37]	X	X	X	-	X	-	X	X	-	X	X	X	X	X
Dorri et al. [38]	X	X	X	P	X	-	X	X	P	X	X	X	X	X
Han et al. [39]	X	X	X	-	X	-	X	X	-	-	X	-	-	-
Lee et al. [40]	X	X	X	-	X	-	X	X	P	P	X	X	X	X
Lin et al. [41]	X	X	X	X	X	P	X	X	-	-	X	X	X	X
Mohanty et al. [42]	X	X	X	X	X	P	X	X	P	P	X	X	X	X
Singh et al. [43]	X	X	X	X	X	-	X	X	X	-	X	X	X	X
Tantidham et al. [44]	X	X	X	-	X	-	X	X	-	P	X	-	X	X
Xu et al. [45]	X	X	X	-	X	-	X	X	-	X	X	X	X	X
Xue et al. [46]	X	X	X	X	X	X	X	X	P	X	X	X	X	X
Zhou et al. [47]	X	X	X	X	X	-	X	X	-	X	X	-	-	-
Zhu et al. [48]	X	X	X	-	X	-	X	X	-	X	X	X	X	X

1 = Identity Holder Consent, 2 = User Control 3 = Provability, 4 = Pseudonymity,
5 = Persistence, 6 = Key Recovery, 7 = Data Protection, 8 = Transparency, 9 = Portability,
10 = Interoperability, 11 = IoT Support, 12 = Scalability, 13 = Data Minimization,
14 = Transaction Costs

X = Criteria is completely fulfilled, P = Criteria is partially fulfilled,
- = Criteria not fulfilled / not applicable

proposed solution [27]. *iv) Pseudonymity* has been addressed by [22] and [33]. Li et al. [22], and Zhang et al. [33] are two examples of how the proposed solutions achieve the identity holder's anonymity. In [22], sensitive data is protected by incorporating cryptographic algorithms and specific file storage functions. In total, eight works incorporate IoT and blockchain to propose their IdMs. Studies by Dwivedi et al. [16], Griggs et al. [17], Uddin et al. [30] already incorporate IoT devices and medical sensors within their research approach. The research conducted by Griggs et al. [17], and Vora [31] only proposed a framework. Therefore, the performance criteria are not applicable. No comparison with the other research papers can easily be made without assessing the framework's feasibility or the transaction costs triggered by a potential implementation.

C. Related Work Focuses on Smart Home

We have selected published articles since 2017 for the comparative study, and Table II presents these studies. Similar to healthcare applications, all the selected works fulfil the same six criteria. None of the proposed solutions completely meets all the defined evaluation criteria. Still, studies by Mohanty et al. [42], and Xue et al. [46] at least fulfil all of them partially. A lightweight blockchain-based solution for identity management regarding various IoT scenarios has been proposed by [42], whereas in [46], a private blockchain-based access control scheme for smart homes is proposed.

It is interesting noting that nearly all selected related works either provide a coherent simulation or an implementation of the proposed solution. Therefore, criteria related to performance, such as *xii) scalability*, *xiii) data minimization*, and *xiv) transaction costs* could be assessed. All the examined smart home-related works consider IoTs as well. Therefore, the evaluation criteria *xi) IoT support* is fulfilled entirely by all the research works. A majority of the selected studies addressed (at least partially) *ix) portability*, and *x) interoperability* criteria regarding the usability of their solutions. The solution proposed by Aggarwal et al. [35] fulfilled both criteria to the full extent, as the blockchain-based IdM for smart homes is entirely integrated within a smart grid ecosystem. Here, homeowners can leverage the solution for transparent and secure energy trading. More research has to be conducted in the smart home domain because the criteria *vi) key recovery*, which represented an enhanced security feature, has only been considered moderately.

III. ChainDiscipline: Proposed Framework

A. Framework

A generic identity management framework called ChainDiscipline has been introduced in Figure 2. This framework should be implemented by a service provider who will be in talks with the user or subscriber (patients, smart homeowners) mainly: *i)* to implement the customised privacy preserving rules, *ii)* to select data storage and data access policy. It is worth noting that data storage policy can include where the data should not be stored, and data access policy can define who (such as third parties) should get access and to which extent (access levels). In Subsection IV-B, we have presented how the proposed blockchain-IoT-based framework has been prototyped based on healthcare and smart homes use cases.

From Figure 2, we can see that the framework consists of three stages: collect, store and use. IoT devices collect data, blockchain platforms (mentioned as HLF[2]) store the on-chain data, while cloud-based (off-chain) storage is used to hold actual user data. We have used Hyperledger Fabric (HLF) [49] for framework implementation purposes. We have selected permissioned HLF for four reasons. They are: *i)* permissioned network is not easily accessible by outsides. HLF is maintained by a network admin, who performs the user verification process before delegating access to the network, *ii)* HLF channel offers privacy inside the blockchain network itself, *iii)* when a single service provider operates a blockchain-based solution (like this case), focusing on throughput, a fully byzantine fault-tolerant consensus might be expensive compared to a crash fault-tolerant consensus protocol. Overall, HLF does not support cryptocurrency and does not require any mining. Without cryptographic mining, blockchain platforms can be deployed at a lower cost than mining-based platforms. *iv)* Although the HLF platform is challenging to work with, but HLF has a higher implementation success rate in many applications [50]. In general, blockchain suffers from scalability problems. Hence, storing unnecessary data in a blockchain network can reduce performance, mainly throughput. To counter the scalability issue, the blockchain will not hold the data but the hashes of the data (metadata) in our framework. It is computationally infeasible to compute the correct input given a hashing output by someone. Cloud storage will host the actual data.

B. Process Flow

As a first step, the user registers him-/her-self as an identity holder at the certificate authority (refer to Figure 2). It is also possible to use national digital IDs. Such digital IDs are issued to citizens to access various public and private web services. After completion of registration, credentials are issued (similar to a unique social security number). The user can now confirm or deny incoming user data access requests. Once IoT devices are configured, the collected data is sent to the Application Program Interface (API) endpoint and evaluated. After successful evaluation, data is processed and submitted to the blockchain using the application gateway. Required metadata is selected as per encoded smart contract rules (set by the user) and later stored encrypted in the list of transactions. New data is appended (later cannot be deleted) to the user data block on the blockchain. Thus, the data block contains the transaction metadata and access logs. In an off-chain (cloud) storage, mainly large files (such as X-Ray images for health records) and access logs are stored. Finally, third parties can access user data based on access credentials.

A QR code is generated for the third parties (refer to Figure 6) to provide access to user data (both for the patient and smart homeowner). A web-service access request is generated after scanning the QR code. During the request phase, the owner of the data gets an alert. If the user denies the

[2]Hyperledger Fabric

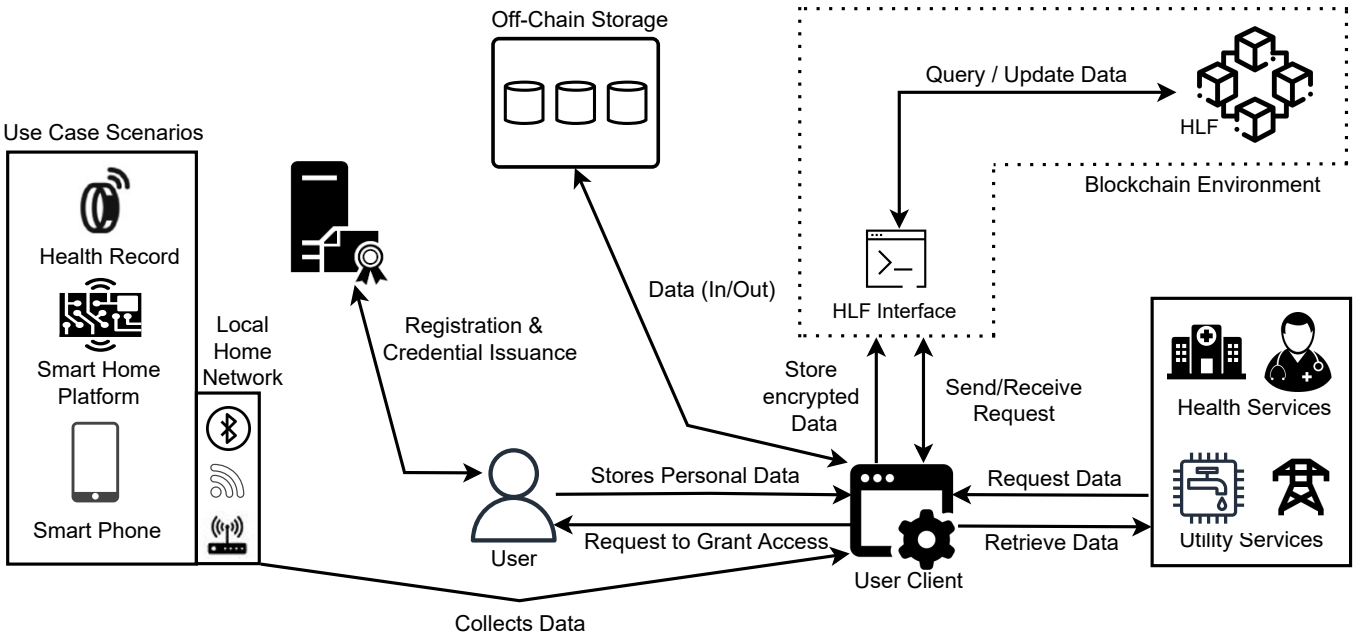

Figure 2. Proposed blockchain-IoT-based identity management framework.

access request, the requested data will not be shared with the requester. Otherwise, once permission is granted, the request will be redirected to the blockchain, and the data will be shared with a third party based on their access privilege (already set by the user). We have denoted third parties as externals during implementation. IoT devices may include wearable medical devices with functionalities (such as electrocardiograms or smart inhalers) that constantly collect data based on the user's activities. Besides the patients, the user group contains one admin to approve membership requests for new users, revoking or declining them. Groups of third parties or externals can include hospitals, pharmacies, medical personnel (e.g., doctors or practitioners), health insurance companies and utility service providers. For instance, medical personnel can obtain medical data only to monitor the patient[3]. Similarly, accessing user-specific data allows health insurance companies to provide more tailored insurance offers. In a smart home scenario, the external can consist of family members, close friends or co-workers. Additionally, security companies and home insurance companies can also be potential externals.

C. New User and IoT Device Management

First, the admin (ideally belonging to the service provider) logs into the system. Once successfully logged in, an overview of all users and their respective statuses is shown. If necessary, the admin is provided with the option to use his/her rights to change a user's status. Thus, the admin sees all the users registered within ChainDiscipline, including their configured IoT devices. To be part of ChainDiscipline, the potential user has to initiate the sign up process. Once the admin approves the request, the new user receives a notification and can login to the system. After successful login, the user is presented with an option to manage all the associated IoT devices linked to the user profile. Management includes registering new IoT devices, accessing the already collected data (as graphs) and managing the IoTs by either updating their details or excluding them.

Following an application scenario, a signed up user can see all registered and past IoT devices and individualise the combination of active IoTs. Additionally, one of the key functionalities for the user is represented by the possibility of sharing data with third parties. In the proposed prototype, the user generates a shareable QR code using a web form that collects information regarding the third-party entity. Correspondingly, the newly generated QR code grants access to the data collected from a specific device within a defined time slot and an expiration date specified during the external registration process. The default configuration for the time frame is set to fourteen days, but it can be easily adjusted using the dropdown option. Thus, a user can share data collected via one of the IoT devices with an external. Lastly, users can also access historically emitted access grants and shared data and external access-related metadata (such as access time, IoT device type, and even the time spent viewing the data could be included).

From the external perspective, there is a limited number of possible actions. The QR code provided by the user needs to be scanned or accessed by a link which enables immediate access to user data. Once the QR code is scanned, the metadata is collected, consisting of the access date and time, the IP address and the device type (e.g. smartphone or laptop). This process

[3]Another potential access request within the healthcare scenario is the data access provided to educational institutions as part of a research study. Such data can be collected via an IoT-based oxygen mask to investigate acute respiratory diseases and later can be used to make an important contribution to clinical trials. It can be crucial in uncertain conditions (such as the COVID-19 pandemic).

ensures after user data is presented, data copy cannot be done.

D. UML Sequence Diagram

Figure 3 presents a UML sequence diagram to show the interactions between the different entities within the framework. The interactions, including the backend routes which use the application gateway to reach the blockchain network, are explained.

1) User Registration: As illustrated in Figure 3, registering a new user is carried out through `RegisterUser`. A user fills up the registration form (on the frontend), which triggers a route `/user/register` to the backend. Next, the backend uses the application gateway to submit the transaction to be processed by the chaincode[4] deployed on the HLF channel `users`. In HLF, channels provide a private communication between specific users. Channels are hidden from other network members. Each channel consists of a separate ledger which can only be read and written to by the related channel members. Overall, the channel offers enhanced privacy inside the network. Regardless of whether the request is valid, the chaincode will respond to the backend and redirect the response to the backend that notifies the user. In this step, one potential reason for encountering an error is that if the desired user already exists, the user will receive an error message. The `UserStatus` operation is mapped as a loop. It means that the registration request is now pending an answer from the admin. Aligned with the `/user/status` route, the application gateway responds to requests on this route, evaluating the `checkUserStatus` function using the chaincode, returning the `userID` if valid, or an error message if the inserted user does not exist within the system. The backend sends a notification to the user again, either with a success or failure message.

The admin evaluates pending user registration requests. In accordance with the `/admin/user/approve` route, the application gateway is used by the backend to handle membership requests by returning a confirmation message. Lastly, `User Login` transaction is consistent with the path `/user/login` and describes the authentication process performed using the chaincode deployed on the `users` channel. The transaction `UserLogin` is evaluated, and if the password and username are correct, the user is authenticated. The user profile management route `/user/update` enables the user to update their personal information, and the backend is responsible for processing such requests and submitting the transaction using the application gateway.

2) IoT Device Management: The 'Add Device' activity is defined within chaincode and can be reached using the route `/user/devices/add`. The transaction `AddDevice` is submitted to the ledger using the chaincode deployed on the channel `devices`. The response received from the execution of this transaction is sent to the backend and is displayed for the user on the frontend. The continuous streaming of data is represented in `Device Stream` loop. Initiated by the device entity and managed using the `/device/stream` route, the chaincode deployed on the channel `streams` processes and forwards transaction `adddevicedata` to be added to the ledger. If the transaction is successful, the user can see that the newly added device is ready to be configured. Once configured, the data stream reaches the backend to add transactions to the ledger.

3) External Management: Finally, the external sequence is covered in the `/external/getuser` route. As a first step, the transaction `CheckExternal` is executed within the chaincode deployed on the channel `externals`. If the transaction is evaluated and confirmed, the following transaction `GetUser` is submitted and processed by a chaincode called externals. In the case of confirmation, `userId` is returned. Subsequently, the execution of the transaction `ReadDevice` returns the relevant information stored on the ledger. To finalise, `GetStreamData` transaction is processed by the device-stream chaincode. Only if all the described transactions are successful, the external will be able to access the data s-/he is authorised to.

IV. Prototyping

This section covers the environment setup, use case development, unit tests results and applicability of this framework. They are explained in the below sub-sections.

A. Environment Setup

The system environment consists of the blockchain network and multiple IoT devices. We have built a four-node Hyperledger Fabric network covering three countries (refer to Figure 4). We have also used a smartwatch and five sensors to support two use-case scenarios.

1) Network Setup: We selected HLF platform[5] to be deployed on four virtual machines (VMs) running Ubuntu (v 20.04.4 LTS). Our multi-node setup increased the reliability and potential scalability of the HLF network. Out of four VMs, one VM is deployed in the Amazon cloud (hosted in Frankfurt and called a host node), and the rest were deployed in a remote cluster (in Romania). The four-node test bed was set up on an eight-core i7 CPU (with 3.60GHz clock speed) and 32 GB of RAM. It was connected to a Cisco Catalyst 3750 series switch which is employed on a Gigabit fibre network connection. The host node runs Linux and initiates the entire network setup process. Each node within this setup is responsible for the orderer organization and a defined set of peer nodes. The cloud-based node-1 (host node) manages the orderer-1 organization and the peer nodes peer0.org1. The remaining nodes, which are hosted in a remote cluster, follow the same logic. Node-2 enables orderer2 organization and hosts the peer nodes peer1.org1, whereas node-3 supplies orderer3 organization and the peer nodes peer0.org2. Lastly, node-4 provides orderer4 organization and peer1.org2 peer nodes. All four nodes have their static IP addresses. One custom bash script is deployed inside a Raspberry Pi 4 (model B) to have an automated node recovery. In case of a node failure, the bash script will run and set the LAN automatically to the predefined list of MAC addresses belonging to the

[4]Hyperledger Fabric termed smart contract as chaincode.

[5]with version 2.2.2

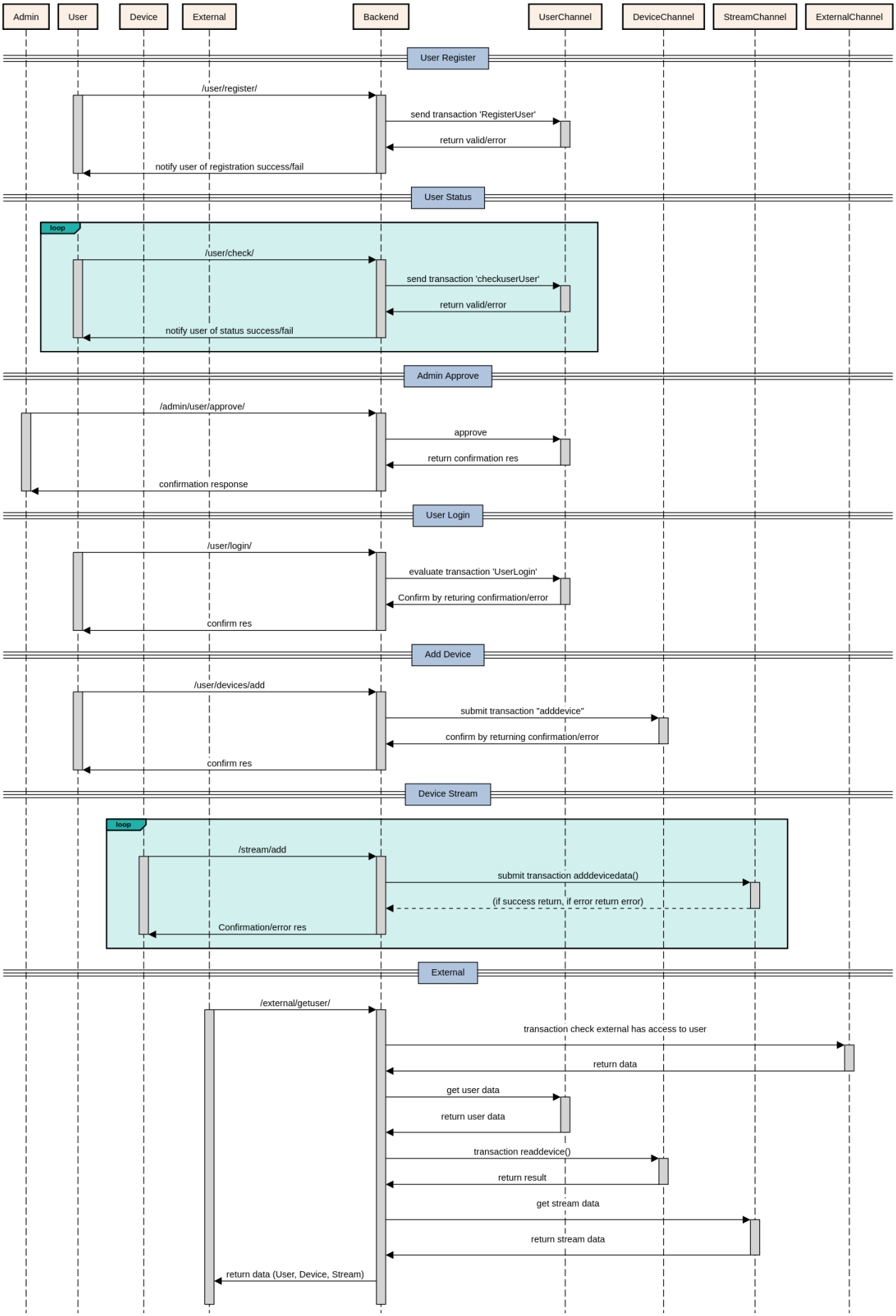

Figure 3. UML sequence diagram covering user registration, IoT device and external management.

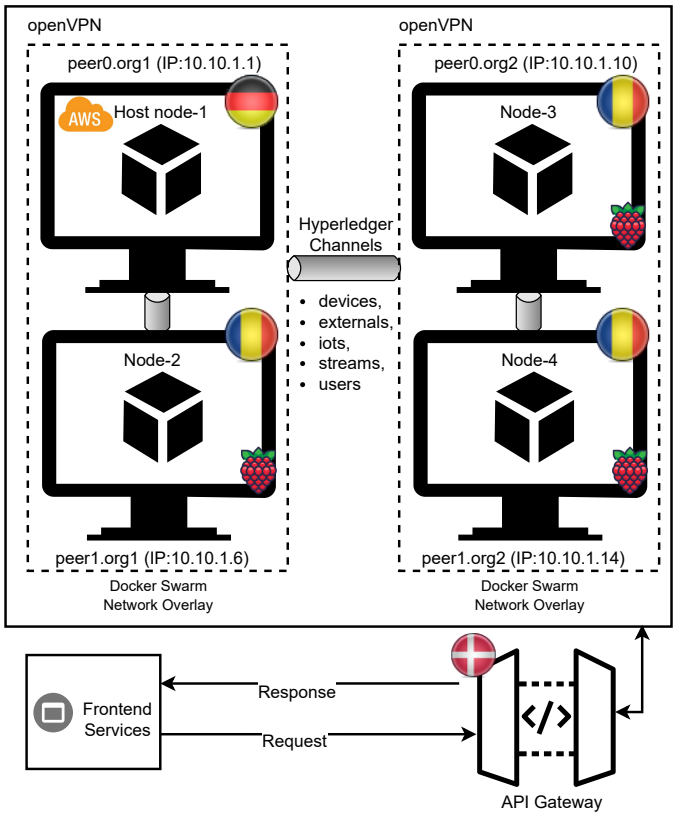

Figure 4. Four node Hyperledger Fabric network setup covering public cloud service (Germany) and on-premise cluster (Romania) and hosted web-services (Denmark).

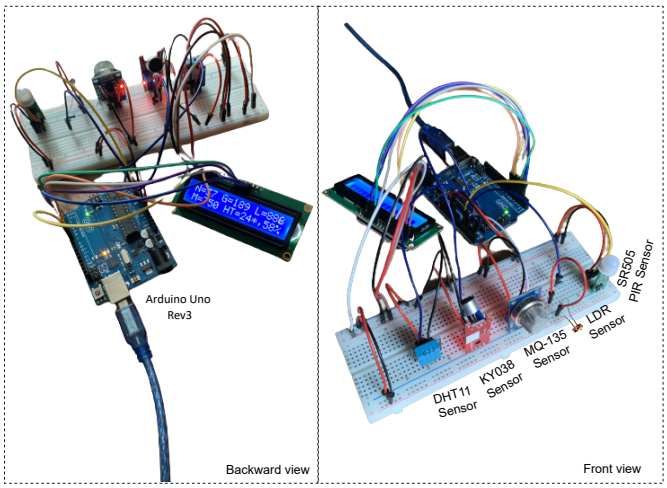

Figure 5. IoT setup with five sensors for smart home use case.

Ubuntu VMs. It is worth noting that losing the connection to a single node does not lead to a forced shutdown of the entire network (as the remaining nodes are interceptable).

2) IoT Setup: The IoT devices aim to stream the data that mimics the functionality of smart home devices for the prototype purpose (refer to Figure 5). The simulation environment of IoT consists of an Arduino Uno Rev3 and five sensors. The sensors are for light (LDR Sensor), motion (SR505 PIR Sensor), gases (MQ-135 Sensor), audio (KY038 Sensor) and temperature, including humidity (DHT11 Sensor). These sensors are used to cover smart home use case scenarios. We have used a commercial smartwatch to collect health-related data.

Overall, we have combined these devices with our customized scripts to simulate both scenarios as realistically as possible. An IoT device supporting software module is developed using JavaScript with a deployment on the Google scripts platform. However, there is a limit on the number of triggers that can be executed per hour per day. To counter the limitation, we use the available network setup of four nodes and a Raspberry Pi. We simulated the events of connecting ten users per host, translating to nearly one hundred users streaming data simultaneously. Still, the platform enabled the deployment of this script with a specific and customizable trigger, which additionally allowed scheduling. The developed software module can receive data from authorized IoT devices and provide the necessary data to demonstrate the required functionalities for the prototyping. Next, a Python script is written to take the user ID and the desired delay (between the specific requests running within a loop) as input. The data is forwarded to the blockchain application gateway. Finally, the transactions are submitted to the ledger. The Hyperledger Explorer provided the first evidence of a constantly increasing number of transactions and a spike in the rate of the number of blocks created per minute (refer to left side of Figure 7).

B. Use Case Scenario

We have selected healthcare and smart home management as two use cases to demonstrate the generability of our proposed blockchain IoT-based identity management framework.

1) Channels: HLF offers channels to support private tunnels of communications. For better data security, one channel can be created where a specific ledger is created, and participants can read the data as per their access level. Five channels have been created to support both use cases. They are `devices`, `externals`, `iots`, `streams`, and `users` (refer to Figure 4). We also define specific functionalities through chaincode customization. The channel `devices` handles the device management, mapping the owner with the device ids. The channel *externals* handles the communication related to externals (or third-party or utility service providers). It primarily consists of information related to entities they belong to, their customer id (meaning user id) and access validity. Access validity has been added to control information accessed by externals. Once the access duration expires, the externals must request new access. It is also worth noting that, all the time, a user has to approve an access request (irrespective of the access duration) of externals. The channel `iots` and channel `users` are primary for smart home use cases, while the same channel `users`, and `streams` build the case for health record management. The channel `iots` handles all the IoT sensors with their readings and associated device-ids. Next, channel `users` holds all relevant information about the user, including the insurance. The insurance is applicable both for the health institutes and also for other externals. Finally, the channel `streams` manages some health-related data (such

as heart rate, oxygen level, sleep records, and stress level, to name a few). We have implemented it in a separate channel to achieve better security.

C. Application Deployment

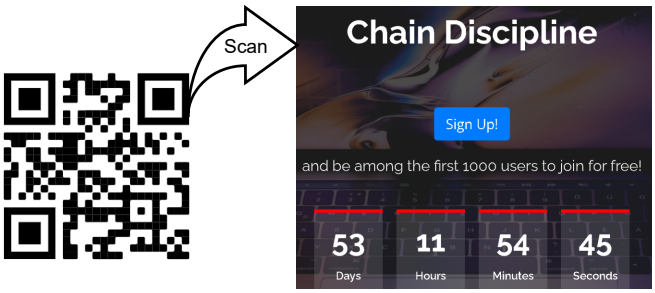

Figure 6. Implemented QR code for chaindiscipline.com.

For testing purpose, we have temporarily deployed the prototype application with the domain name chaindiscipline.com. However, the primary aim of this work is to demonstrate the feasibility of such a framework and test the seamless connection of the frontend with the backend (for more, refer to Table IV). We also have demonstrated IoT integration with blockchain. The frontend part can manage connected IoT devices, and the application gateway routes access requests from external/third parties. The externals scan the QR code to get necessary user data access. Using case-specific users, IoT devices and externals are mapped by their respective channels on the HLF network. Chaincodes have also been customized for required support. In our framework, the user must authorize all data access requests, and only the necessary amount of information will be sent to the requestor. It highlights the privacy-preserving nature of the developed prototype. A service provider should implement the framework where a subscription-based business model can be applied (detailed discussion is in Sub-section IV-E). In Figure 6, we showed our custom QR code for easy access to the solution. The uni-directional information flow enabled the implementation of the QR code. The code generation feature provides user data access to an external with the help of the application gateway. We have used Axios[6] client to bridge the backend connection with the frontend. Primarily, the landing page, user registration, and login page are implemented using Axios. Table III shows the main domain chaindiscipline.com and all its sub-domains. The second column of the table further provides a brief description. Overall, users communicate with the frontend and backend via an API gateway (refer to Figure 4). This API gateway handles the requests from frontend services (including QR code). User credentials are stored in the `users` channel. Apart from adding login and logout functions, other relevant functionalities can be added to the chaincode. During a function call, a respective route is accessed on the backend, which forwards the call (contains user input) to the login function on the application gateway. If the user-provided credentials are matched to those stored on the ledger,

[6]https://axios-http.com/

Table III
IMPLEMENTED CLIENTS DESCRIPTIONS

Domain name	Destination
chaindiscipline.com	Accessing the landing page
app.chaindiscipline.com	Provide access to frontend
api.chaindiscipline.com	Provide access to backend RESTful API
explorer.chaindiscipline.com	Provide access to Hyperledger Explorer
zabbix.chaindiscipline.com	For infrastructure monitoring

the user can login. During the implementation of this login functionality, an additional security measure (called *session*) is implemented to prevent cross-site request forgery attacks or stolen cookie attacks. Apart from that, the session also enhances performance and provides a better user experience. For instance, once logged in, the user does not need to login again when re-visiting an open tab within the application; as long as the user did not log out, there is no need to login until the session expires.

Customized routes are developed to process requests (such as user registration, login, or updating existing users), and chaincode is also extended for the required support. A call to `init ledger` function is made to submit the transaction during the initialization of `users` channel. It contains a predefined list of users (as a JSON object). The testing and subsequent evaluation revealed the ability of the prototype to fetch existing users and the successful execution of other functionalities (such as user registration). One by one, the channels for the IoT devices and the externals are created. Next, the chaincode is customized to match the required functionalities, and finally, the deployment was carried out successfully. Accordingly, the routes for the application gateway are also implemented. Hyperledger Explorer has been used for real-time monitoring of transactions, including block creations within the network (refer to Figure 7).

The interactive dashboard (refer to (left side) of Figure 7) contains some key numbers of the deployed blockchain network. Blocks and transactions can be monitored in an interactive tool based on the selected time frame (per minute or per hour). The left side of Figure 7 illustrates the implementation using a four-node setup that includes information related to chaincode. It shows the data related to channel `streams`. It listed (on top) the total number of blocks created (i.e. 9072), and total successful transactions submitted (i.e. 43476). Information is also illustrated via a pie chart that includes the names of the peers and transaction details by the organizations. A trace of the created blocks with timestamps, including the channel and the associated hashes, wraps up the dashboard. Next, the right side of Figure 7 shows all submitted transactions (with other details) that are based on a defined timeframe (i.e. from May 1^{st} to May 16^{th}). Other details include the creator (organization and specific channel) and the hash value of the transaction ID. Furthermore, the type of transaction includes a chaincode that has triggered the respective transaction and a timestamp. This part of the monitoring allows us to trace any submitted transaction. During the testing and evaluation, including the `init ledger` for each chaincode has been successfully deployed. It ensures that

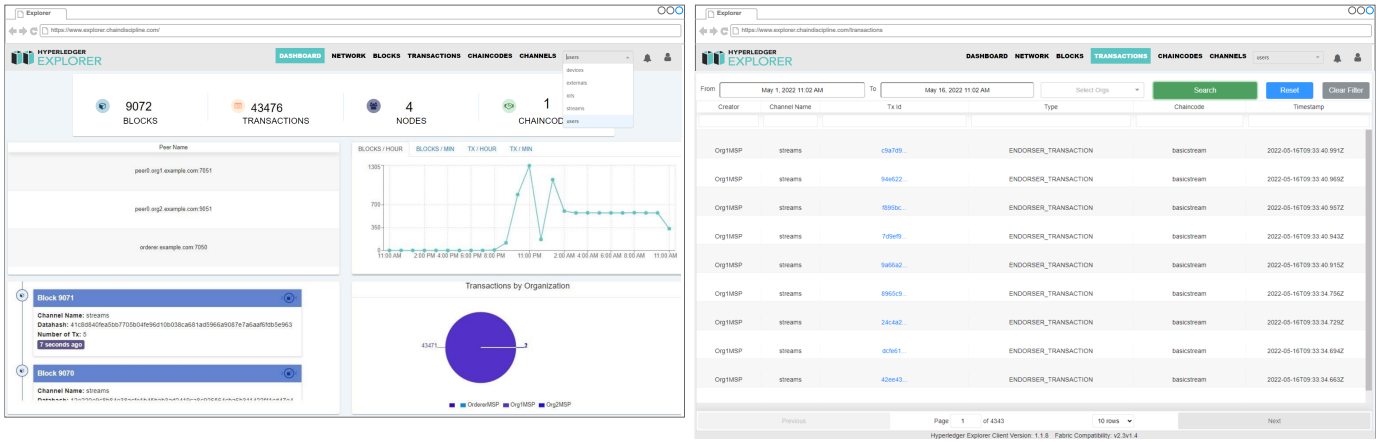

Figure 7. Hyperledger Explorer dashboard shows overall transactions and other network details are listed on the left side and channel `streams` specific transaction updates are also visible on the right side.

users, connected IoT devices, and externals are registered to the network. It enables us to conduct tests by sending queries through the command-line interface. Testing the deployed chaincode on the connected channel and executing the queries are very important, as it is the only way to ensure that chaincode functionality is satisfactory.

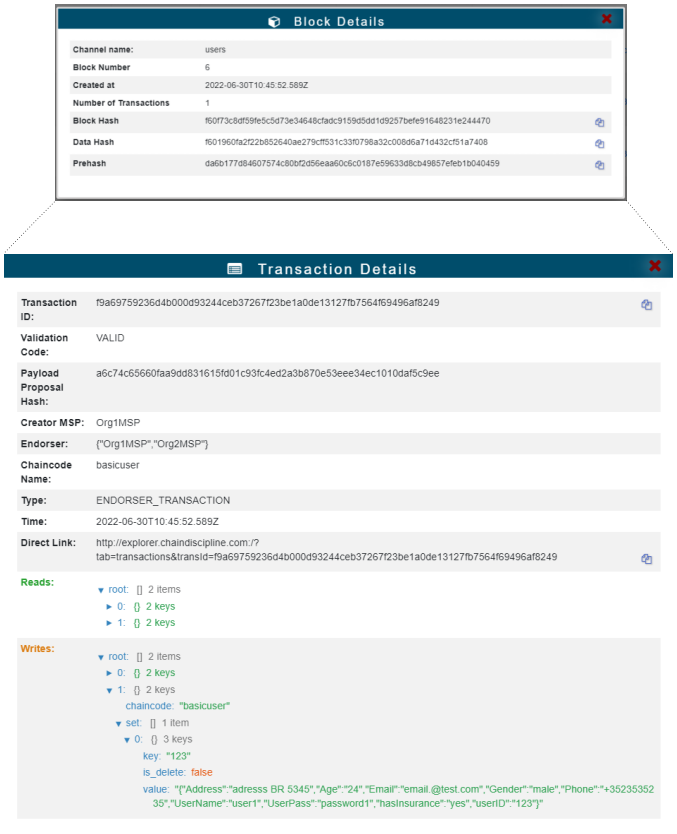

Figure 8. Transaction details of a user block.

1) Block and Transaction Level Details: Here, we present the transactions and associated block details using Hyperledger Explorer related to healthcare, and smart home data management use cases. Each block details include information related to the channel name, block number, number of transactions and

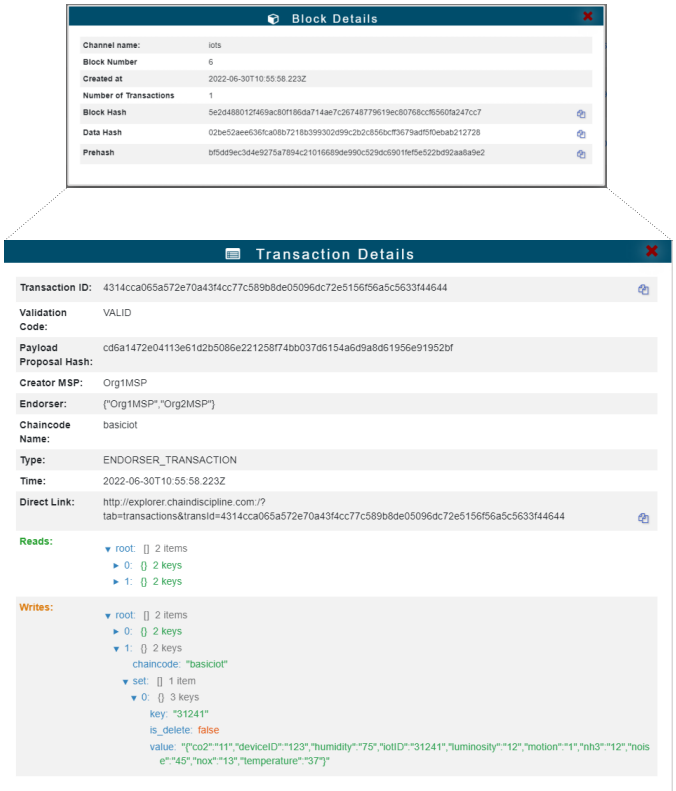

Figure 9. Details of a sample IoT block to support smart home.

hash value. Next, the transaction details include a transaction overview, a direct web link, and the details of the read-write operations. Figure 8 presents a sample of the block and transaction details related to the user record. It is worth noting that user block is common in both use cases. Collected user details are always implementation dependent. It is also possible to have a single channel that combines user and iot block for smart home use case. Similarly, combine user and stream records for health data management use case. For the modular approach, we decided to separate them. Figure 9 presents the transaction details related to IoT devices. These

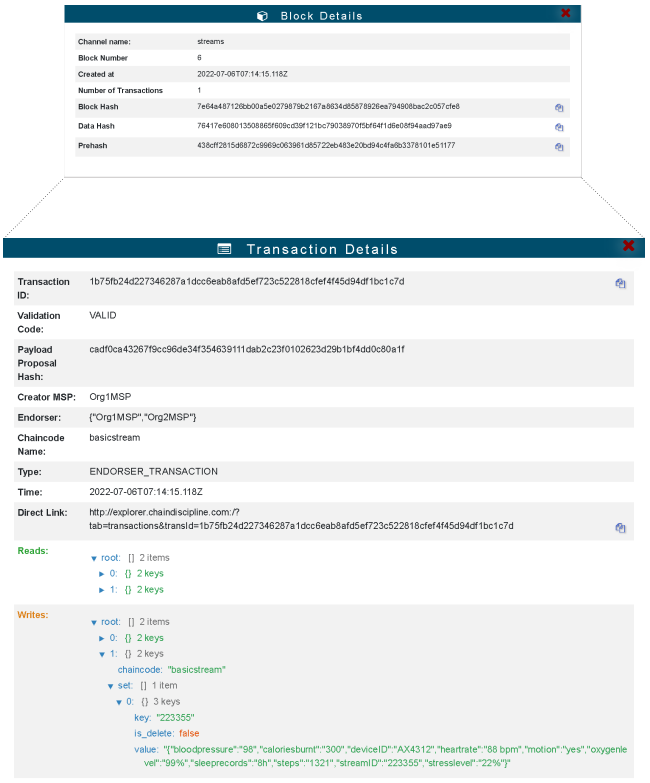

Figure 10. Details of a sample block to support health record.

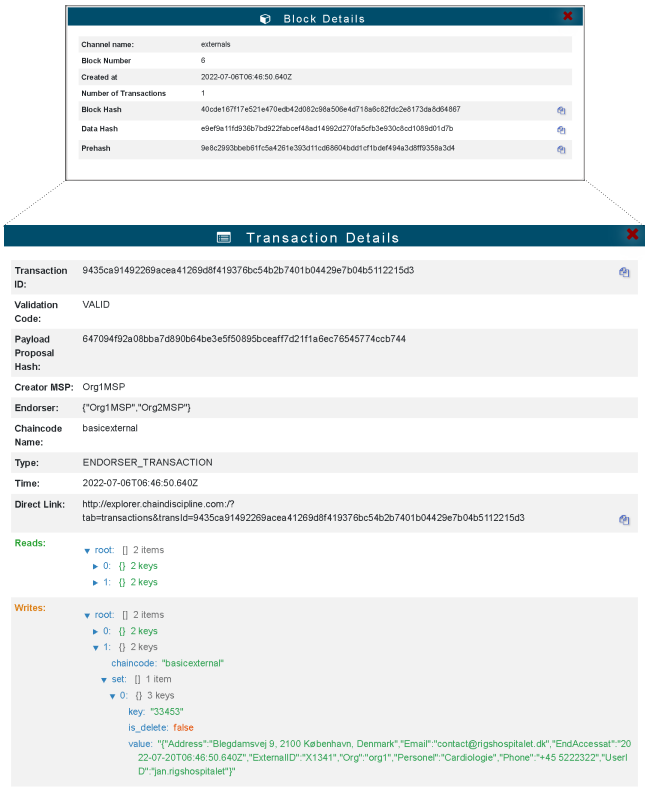

Figure 11. Details of a sample block to support externals.

data are collected using our five sensors. The frequency of data collection is again customisable. Figure 10 presents the transaction details related to health data. These data are collected using a commercially available smartwatch. We further customised the parameters of the smartwatch to collect health data. For better readings, maybe medical devices can be used. Figure 11 presents the transaction details related to an external (hospital in this case). Similar to hospitals, electric or other utility suppliers can also be added. An external entity has to provide details while generating a registration request (via a web form). The information will be sent to the owner of the health record. It can also be seen that in the record, there is an entry called `EndAccessat` which sets the validity of the external entity. After the expiry, a new registration process has to be initiated. It can be seen that multiple privacy-preserving and data-control features can be added to restrict data access to externals further.

D. Unit Testing Results

To demonstrate the overall functional suitability of our solution, basic unit tests are done on the final version of the prototype. Thirteen tests are made, and the expected and actual response values are presented. The end-to-end test results are displayed in Table IV. It can be seen from the results that six main functions and multiple sub-functions are tested. If a function returns the exact response, the test is successful. For instance, during `User Registration` function test, the route `users/[POST]register` is tested. Since the received response is 200, the determined route is functioning. The 'Actual Output' indicates if the tests are passed or not. Similarly, we have completed unit tests for *Restricted Access* scenarios. In these test cases, the expected server response is set to 501, which refers to unauthorized access. The test indicates whether or not an unauthorized function (such as adding a device by a non-registered user) is triggered.

E. Applicability

There is a great possibility to publish different versions of chaincode with additional privacy-preserving features by applying a subscription-based business model. For example, a free version of the solution can allow a small number of privacy-preserving functions, which a user can execute and further include a limited number of IoT devices. The premium version can have additional functions (e.g. enable the user to connect as many IoT devices as possible). The premium-plus version can incorporate all the premium plan features but also allow other features (such as letting the user add family members or friends as users or having the right to propose a new entity as an additional external) based on the application domain. The monthly fee can accordingly differ as per subscribed plans. Another option can be the freemium subscription model, which means that the basic version would be at no cost to the customer, but as it lacks several functionalities, the paid versions may be more interesting for other customers who need more sophisticated products or services. A study has already been conducted to investigate the value propositions of such SSI solution and the impacts on its adoption [51] and in another study, SSI is included in the context of IoT-as-a-service to create new

Table IV
END-TO-END UNIT TEST RESULTS TOGETHER WITH TESTED ROUTES

Test Case (Route)	Expected Output	Actual Output
1. User Registration (`users/[POST]register`)		
1.1. User clicks the sign up button on landing page 1.2. User fills out registration form with defined credentials	Landing page displayed Sign up done	Landing page displayed Sign up done
2. Admin Login (`admin/[POST]admin login`)		
2.1. Admin enters correct credentials 2.2. Admin enters wrong credentials or nothing at all	Admin page displayed Sign in denied, login page displayed	Admin page displayed Sign in denied, login page displayed
3. Admin Approve (`admin/[POST]approve user`)		
3.1. Admin clicks on 'Approve' button for any solicited new user registration	New user registered successfully	New user registered successfully
4. User Sign In (`users/[POST]signin`)		
4.1. User enters correct credentials 4.2. User enters wrong credentials or nothing at all	User page displayed Sign in denied, login page displayed	User page displayed Sign in denied, login page displayed
5. User Functions (`users/[POST]generate qrcode`)		
5.1 Generate QR code 5.2. Add a new Device 5.3 Add new Stream	QR code generated successfully Device added successfully Stream added successfully	QR code generated successfully Device added successfully Stream added successfully
6. Restricted Access (`device/[POST]add device`)		
6.1. Unauthorised Device add 6.1. Unauthorised Device update 6.2. Unauthorised Device delete	Device registration fail Device update not possible Device deletion not possible	Device registration fail Device update not possible Device deletion not possible

business opportunities [52]. Thus, the SSI research community needs to define a consistent narrative and a user-focused data ecosystem to increase the chances of widespread adoption. Better implementation of such a framework may also improve the health conditions of marginalised groups where medical resources are very scarce.

V. DISCUSSION AND LIMITATIONS

There is a need to build a robust SSI ecosystem and related services as online user data are collected, stored and processed as 'normal' data by many online entities. A theoretical framework for an identity metasystem proposes seven laws [53], and later, these laws also influence the SSI concept. These seven laws are *i) User control and consent:* is to control the identity holder over the identity-related information and whom to share with, *ii) law of minimal disclosure:* allow the least required amount of information to be delegated. The author further places emphasis on the user control via his/her *iii) justifiable parties' law:* which only allows required parties to access the data, while *iv) directed identity:* emphasises keeping single connections between entities. Next, *v) pluralism of operators and technologies:* enables an identity ecosystem across multiple technologies, run by multiple providers, where *vi) human integration law:* aim to improve the usability of the identity management solution by keeping the user as a central component and lastly, *vii) consistent experience across contexts:* aims to offer a simple and consistent user experience. Later, ten guiding principles for self-sovereign identity have also been proposed in [54]. The list of guiding principles has been expanded by adding 'Provability', which means all the claims made must be verifiable [55]. Below, Table V summarises these ten principles and indicates whether our blockchain-IoT-based framework supports them.

Table V
SUPPORT TO SELF-SOVEREIGN IDENTITY PRINCIPLES [54]

SSI Principles	Supporting
Existence: For each user an independent identity exists	Yes
Control: Users must have full control over his/her identities	Yes
Access: Users must be able to access his/her data	Yes
Transparency: Systems and algorithms must be easy to understand, and free	Yes
Persistence: Identities must be long-lived	Yes
Portability: Identity related services and information must be transportable	Yes
Interoperability: Identities should be widely usable, even being global identity	Yes
Consent: Users must give their consent to share their identities with other parties	Yes
Minimalization: Only minimal amount of data necessary for disclosing claims is used	Yes
Protection: Rights and freedom of the users must be protected in a conflict	Yes

It can be seen that our framework supports all ten privacy-preserving principles. Secure self-sovereign identity management systems can protect social security information or financial information (such as credit card details). The introduction of privacy rules (such as the right to be forgotten) by concerned countries can also influence how and where user identity-related data should be stored. Implementing the right to be forgotten is not trivial, especially in blockchain-based solutions, where data can only be appended. In our framework, we have used off-chain storage where data can be removed. There is no universal answer to the country-specific privacy rules. Thus, the solution has to be customized to satisfy the law of the land. This paper proposes a generic blockchain-IoT-based identity management framework to improve online user

identity, where country-specific identity-preserving rules can be encoded into the chaincode. Blockchain brings potential benefits (such as very high data tampering resistance features for identity holders and enhanced data security). With this developed prototype, a first step towards building a more sophisticated solution, including relevant security features, has been made. In this framework, the risk of sharing sensitive data is also diminished as only the pre-authorized parties can gain access.

It is also worth noting that the security of user data is the most important issue when designing self-sovereign identity management solutions. Insecure devices (such as blockchain network nodes, smartphones, smartwatches and IoT devices, including security cameras) handling can lead to a single source of vulnerability. However, we foresee three limitations to our existing prototype. They are *i)* no key recovery management feature is provided, *ii)* the solution primarily relies on the QR code-based data request approach, which may reduce the usability of our prototype, and *iii)* the simulation environment should consist of production-grade IoT devices.

VI. Conclusion and Future Work

Managing online user identity is complex but essential in today's complex Internet. User data is being shared without the user's proper consent. The paper proposes and implements a blockchain-IoT-based generic identity management framework. We showed its generability by mapping two real use cases (healthcare and smart home). We also demonstrate its operability and related functionalities via our developed prototype. Unit test reports of primary functionalities are also presented. In this prototype, a QR code-based user data access request (initiated by third parties) mechanism has also been implemented. Apart from solving the three limitations mentioned as future work, we will also improve the verification process of externals (or third parties) for both use cases. Exploring and investigating additional features to improve the user experience will enhance the prototype's overall acceptance and allow those features to be established within the research area.

VII. Contribution Statement

Marius Popa: Conceptualization, Investigation, Methodology, Software, Writing – Original Draft. **Sebastian Michael Stoklossa:** Conceptualization, Investigation, Methodology, Writing – Original Draft. **Somnath Mazumdar:** Conceptualization, Methodology, Investigation, Resources Writing - Review & Editing, Visualization, Supervision.

References

[1] R. Mahajan, D. Wetherall, and T. Anderson, "Negotiation-based Routing between Neighboring ISPs," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, 2005, pp. 29–42.

[2] F. Staff, "A Look At What ISPs Know About You: Examining the Privacy Practices of Six Major Internet Service Providers," Federal Trade Commission, Tech. Rep., 2021.

[3] J. Su, A. Shukla, S. Goel, and A. Narayanan, "De-Anonymizing Web Browsing Data with Social Networks," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1261–1269.

[4] H. Burch, "Tracing Anonymous Packets to their Approximate Source," in *14th Systems Administration Conference*, 2000.

[5] I. Sánchez-Rola, M. Dell'Amico, P. Kotzias, D. Balzarotti, L. Bilge, P.-A. Vervier, and I. Santos, "Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control," in *Proceedings of the ACM Asia Conference on Computer and Communications Security*, 2019, pp. 340–351.

[6] J. Bernal Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. Torres Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," *IEEE Access*, vol. 7, pp. 164 908–164 940, 2019.

[7] R. Soltani, U. T. Nguyen, and A. An, "A survey of self-sovereign identity ecosystem," *Security and Communication Networks*, vol. 2021, 2021.

[8] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020.

[9] P. Dunphy and F. A. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE Security & Privacy Magazine*, vol. 16, no. 4, pp. 20–29, 2018.

[10] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*. IEEE, 2017, pp. 137–141.

[11] M. Benchoufi, R. Porcher, and P. Ravaud, "Blockchain protocols in clinical trials: Transparency and traceability of consent," *F1000Research*, vol. 6, p. 66, 2017.

[12] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller, "Blockchains everywhere - a use-case of blockchains in the pharma supply-chain," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 772–777.

[13] J. Brogan, I. Baskaran, and N. Ramachandran, "Authenticating health activity data using distributed ledger technologies," *Computational and structural biotechnology journal*, vol. 16, pp. 257–266, 2018.

[14] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Generation Computer Systems*, vol. 95, pp. 420–429, 2019.

[15] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, 2018.

[16] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for iot," *Sensors (Basel, Switzerland)*, vol. 19, no. 2, 2019.

[17] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, p. 130, 2018.

[18] Hang, Choi, and Kim, "A novel emr integrity management based on a medical blockchain platform in hospital," *Electronics*, vol. 8, no. 4, p. 467, 2019.

[19] H. R. Hasan, K. Salah, R. Jayaraman, J. Arshad, I. Yaqoob, M. Omar, and S. Ellahham, "Blockchain-based solution for covid-19 digital medical passports and immunity certificates," *IEEE Access*, vol. 8, pp. 222 093–222 108, 2020.

[20] L. Ismail, H. Materwala, and S. Zeadally, "Lightweight blockchain for healthcare," *IEEE Access*, vol. 7, pp. 149 935–149 951, 2019.

[21] H. Kaur, M. A. Alam, R. Jameel, A. K. Mourya, and V. Chang, "A proposed solution and future direction for blockchain-based heterogeneous medicare data in cloud environment," *Journal of medical systems*, vol. 42, no. 8, p. 156, 2018.

[22] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, and S. Liu, "Blockchain-based data preservation system for medical data," *Journal of medical systems*, vol. 42, no. 8, p. 141, 2018.

[23] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–5.

[24] T. Mikula and R. H. Jacobsen, "Identity and access management with blockchain in electronic healthcare records," in *2018 21st Euromicro Conference on Digital System Design (DSD)*. IEEE, 2018, pp. 699–706.

[25] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure ehrs sharing of mobile cloud based e-health systems," *IEEE Access*, vol. 7, pp. 66 792–66 806, 2019.

[26] A. R. Rajput, Q. Li, M. Taleby Ahvanooey, and I. Masood, "Eacms: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84 304–84 317, 2019.

[27] A. Roehrs, C. A. Da Costa, and R. Da Rosa Righi, "Omniphr: A distributed architecture model to integrate personal health records," *Journal of biomedical informatics*, vol. 71, pp. 70–81, 2017.
[28] J. Song, T. Gu, Z. Fang, X. Feng, Y. Ge, H. Fu, P. Hu, and P. Mohapatra, "Blockchain meets covid-19: a framework for contact information sharing and risk notification system," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 269–277.
[29] T. T. Thwin and S. Vasupongayya, "Blockchain-based access control model to preserve privacy for personal health record systems," *Security and Communication Networks*, vol. 2019, pp. 1–15, 2019.
[30] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "Continuous patient monitoring with a patient centric agent: A block architecture," *IEEE Access*, vol. 6, pp. 32 700–32 726, 2018.
[31] J. Vora, A. Nayyar, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, "Bheem: A blockchain-based framework for securing electronic health records," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.
[32] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.
[33] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of medical systems*, vol. 42, no. 8, p. 140, 2018.
[34] X. Zheng, S. Sun, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Meré, "Accelerating health data sharing: A solution based on the internet of things and distributed ledger technologies," *Journal of medical Internet research*, vol. 21, no. 6, p. e13583, 2019.
[35] S. Aggarwal, R. Chaudhary, G. S. Aujla, A. Jindal, A. Dua, and N. Kumar, "Energychain," in *Proceedings of the 1st ACM MobiHoc Workshop on Networking and Cybersecurity for Smart Cities*. New York, NY, USA: ACM, 2018, pp. 1–6.
[36] S. Arif, M. A. Khan, S. U. Rehman, M. A. Kabir, and M. Imran, "Investigating smart home security: Is blockchain the answer?" *IEEE Access*, vol. 8, pp. 117 802–117 816, 2020.
[37] M. A. Bouras, Q. Lu, S. Dhelim, and H. Ning, "A lightweight blockchain-based iot identity management approach," *Future Internet*, vol. 13, no. 2, p. 24, 2021.
[38] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2017, pp. 618–623.
[39] D. Han, H. Kim, and J. Jang, "Blockchain based smart door lock system," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2017, pp. 1165–1167.
[40] Y. Lee, S. Rathore, J. H. Park, and J. H. Park, "A blockchain-based smart home gateway architecture for preventing data forgery," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, 2020.
[41] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K. R. Choo, "Homechain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818–829, 2020.
[42] S. N. Mohanty, K. C. Ramya, S. S. Rani, D. Gupta, K. Shankar, S. K. Lakshmanaprabu, and A. Khanna, "An efficient lightweight integrated blockchain (elib) model for iot security and privacy," *Future Generation Computer Systems*, vol. 102, pp. 1027–1037, 2020.
[43] S. Singh, I.-H. Ra, W. Meng, M. Kaur, and G. H. Cho, "Sh-blockcc: A secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology," *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, p. 155014771984415, 2019.
[44] T. Tantidham and Y. N. Aung, "Emergency service for smart home system using ethereum blockchain: System and architecture," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2019, pp. 888–893.
[45] Q. Xu, Z. He, Z. Li, and M. Xiao, "Building an ethereum-based decentralized smart home system," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018, pp. 1004–1009.
[46] J. Xue, C. Xu, and Y. Zhang, "Private blockchain-based secure access control for smart home systems," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 12, 2018.
[47] Y. Zhou, M. Han, L. Liu, Y. Wang, Y. Liang, and L. Tian, "Improving iot services in smart-home using blockchain smart contract," 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. 81–87.
[48] X. Zhu, Y. Badr, J. Pacheco, and S. Hariri, "Autonomic identity framework for the internet of things," in *2017 International Conference on Cloud and Autonomic Computing (ICCAC)*. IEEE, 2017, pp. 69–79.
[49] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: Association for Computing Machinery, 2018.
[50] N. Vadgama and P. Tasca, "An analysis of blockchain adoption in supply chains between 2010 and 2020," *Frontiers in Blockchain*, vol. 4, p. 610476, 2021.
[51] M. Lockwood, "Exploring value propositions to drive self-sovereign identity adoption," *Frontiers in Blockchain*, vol. 4, 2021.
[52] S. de Diego, C. Regueiro, and G. Macia-Fernandez, "Enabling identity for the iot-as-a-service business model," *IEEE Access*, vol. 9, pp. 159 965–159 975, 2021.
[53] K. Cameron, "The laws of identity," *Microsoft Corp*, vol. 12, pp. 8–11, 2005.
[54] C. Allen, "The path to self-sovereign identity," 2016, last accessed 01-August-2022. [Online]. Available: https://www.coindesk.com/markets/2016/04/27/the-path-to-self-sovereign-identity/
[55] D. van Bokkem, R. Hageman, G. Koning, L. Nguyen, and N. Zarin, "Self-sovereign identity solutions: The necessity of blockchain technology."

Marius Popa was born in Medias, Romania in 1985. He received a B.Sc. degree in Business Economics Information Technology from the University of Southern Denmark in 2020. In 2022 he completed his M.Sc. in Business Administration and Information Systems from Copenhagen Business School, Denmark. His research focus during his Master's was Blockchain, Machine Learning, and IoT. Marius has been working as an account Cloud engineer at Oracle since September 2022.

Sebastian Michael Stoklossa was born in San Ramon, Costa Rica in 1996. He received a B.Sc. degree in Management & Economics with a major in Information Systems and Supply Chain Management from the University of Hohenheim, Germany, in 2019. In 2022, he completed his M.Sc. in Business Administration and Information Systems from Copenhagen Business School, Denmark. His research focus during his Master's was on Innovation Strategies, Machine Learning, and Bigdata Management. Professionally, he works on his career path in Data Analytics.

Somnath Mazumdar Somnath is an Assistant Professor at Copenhagen Business School. His research interests focus on high-performance/throughput computing, performance engineering, blockchain, and machine learning. He holds a PhD in Computing Systems from the University of Siena, Italy, and an M.Sc. in Distributed Computing from Polytech Nice Sophia Antipolis, France. Somnath has also worked for premier European research institutes and contributed to multiple large European and international research projects at various levels.