Towards Blockchain-based Identity and Access Management for Internet of Things in Enterprises

Martin Nuss[1], Alexander Puchta[1], and Michael Kunz[2]

[1] University of Regensburg, Universitätsstraße 1, 93053 Regensburg, Germany
martin.nuss@student.ur.de
alexander.puchta@ur.de
[2] Nexis GmbH, Franz-Meyer-Straße 1, 93053 Regensburg, Germany
michael.kunz@nexis-secure.com

Abstract. With the Internet of Things (IoT) evolving more and more, companies active within this area face new challenges for their Identity and Access Management (IAM). Namely, general security, resource constraint devices, interoperability, and scalability cannot be addressed anymore with traditional measures. Blockchain technology, however, may act as an enabler to overcome those challenges. In this paper, general application areas for blockchain in IAM are described based on recent research work. On this basis, it is discussed how blockchain can address IAM challenges presented by IoT. Finally, a corporate scenario utilizing blockchain-based IAM for IoT is outlined to assess the applicability in practice. The paper shows that private blockchains can be leveraged to design tamper-proof IAM functionality while maintaining scalability regarding the number of clients and transactions. This could be useful for enterprises to prevent single-point-of-failures as well as to enable transparent and secure auditing & monitoring of security-relevant events.

Keywords: Identity and Access Management · Access Control · Blockchain · Internet of Things.

1 Introduction

Identity and Access Management (IAM) has become a highly relevant task for enterprises and organizations in recent years [20]. One major change enterprise IAM must deal with is the concept of Internet of Things (IoT). Haller et al. [12] define the term IoT as the integration of physical objects into information networks. By this means, smart devices can interact with services via the internet and participate actively in business processes [12].
When speaking of IoT, the concept of identity does not only encompass user identities but also extends to IoT devices and services [5]. Secure machine to machine (M2M) communication requires reliable mechanisms to establish trust and access control between IoT devices, data and network resources [1]. The communicating IoT devices must be uniquely identifiable to enable authenticity

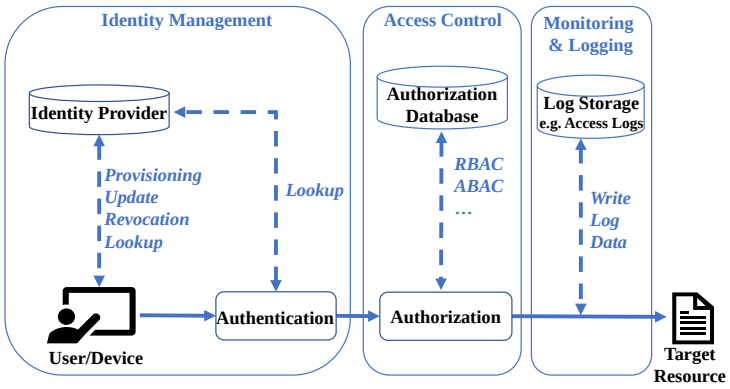

Fig. 1. Overview of IAM functionalities

and to prevent security breaches [3]. In order to achieve this, IAM provides the following three main components. Zhu et al. [33] state that identity provisioning, update, revocation and lookup pose a core set of **Identity Management** operations. The identities of all communicating entities must be secured to prevent identity theft. Data and networks used by IoT devices must be protected by **Access Control** mechanisms to prevent unauthorized access to enterprise resources and confidential IoT data [3]. Finally, IAM incorporates **Monitoring & Logging** functionalities to be able to store and trace critical information in a secure and auditable manner [20]. Figure 1 summarizes these three main functions of IAM.

Various enterprises consider blockchain technology to address IAM challenges in an IoT environment [15]. A blockchain is a distributed database of verifiable records containing transactions which are shared among participating parties. Each transaction is verified through consensus. The records within a blockchain are linked by cryptographic hashes. Each block contains the hash value of the previous block [7]. For a more detailed description please confer chapter 4. Even though several works propose blockchain technology for access control (e.g. [19, 22]), research has put little emphasis on whether and how blockchain can be applied to the more fundamental concept of IAM in enterprises. The question whether the blockchain technology can deal with the mentioned IAM challenges as a whole in context of enterprise IoT has not been entirely addressed in research yet. In this paper, the following research questions will be examined:

1. Which challenges faces IAM within an enterprise IoT environment?
2. Can blockchain technology be used as an enabler for IAM within enterprise IoT and the corresponding challenges?
3. What is a realistic use case for blockchain-based IAM and enterprise IoT?

Our underlying research methodology is shown in Figure 2 and is based on the principles of [13]. In order to achieve the research goals defined above, we firstly

analyze the relevant body of knowledge for IoT and blockchain technology in Section 2 and Section 4.1. We afterwards derive challenges which modern IAM has to face within an enterprise IoT context (1) in Section 3. Thus, we are able to define constraints and requirements for the integration of blockchain approaches later on. Within (2), blockchain-based approaches for enterprise IAM will be presented in Section 4. Furthermore, we show how each component of IAM named above can be supported by blockchain technology. We analyze each component under the aspect of the previously defined challenges. In Section 5, we use the results to create a theoretical use case based on our IAM knowledge and the existing literature (3). It intends to show the applicability of our approach. Finally, within Section 6, advantages and challenges of blockchain usage will be discussed (4).

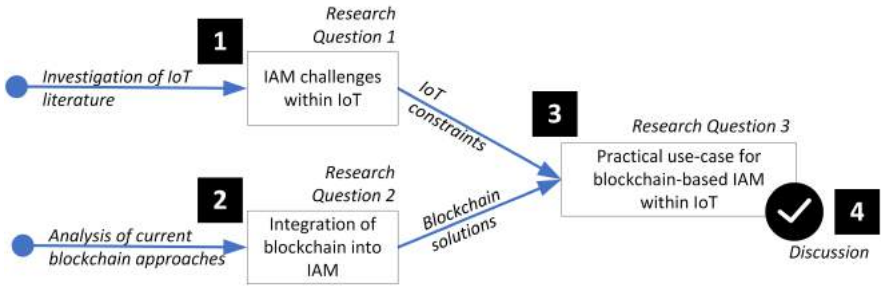

Fig. 2. Methodology

2 Related Work

The topic of IAM, IoT and arising challenges has already been addressed in various research works in recent time.

Adireddy & Gottapu [1] find that IoT devices tend to be unsuitable for intensive operations or large storage due to a lack of computational resource. Zhu et al. [33] state that the large number of different entities which communicate in IoT networks require a robust and extensible structure to enable secure Identity Management. According to Roman et al. [25], a common framework of secure protocols are required to enable governance and interoperability between users and a variety of different devices. Finally, all entities and their identities must be managed in a scalable way in terms of changing space and network requirements over time.

Trnka & Cerny [30] propose an IAM framework which enables device cooperation. The authors leverage a centralized identity store which keeps records of all connected devices with their unique identifiers. The centralized identity store uses role based access control (RBAC) [27] to manage device access. Device accounts and role assignments are created by an administrator. Devices use tokens

for authentication. For confidential communication, devices exchange their tokens and compare them to the tokens stored at the central identity store [30].
Salman et al. [26] suggest the use of a gateway layer and a controller layer between the central store and IoT devices to deal with device heterogeneity. Virtual IPv6 addresses are used as device identifiers. The controller layer provides a public key certificate to each gateway within the network. To authenticate, things use their unique IPv6 identifier together with a nonce that is encrypted with the respective gateway's public key [26].
Gusmeroli et al. [11] propose capability based access control (CapBAC) because it provides deeper granularity as well as easy support for right delegation. Capabilities are objects which are issued by entitled subjects to another subject the capability is granted to. Capabilities must be transmitted to all subjects in the network. To illustrate the decision process, let's assume a situation where an employee Alice requests access temperature data of an engine system employed in an assembly line. The responsible administrator Bob issues a capability to Alice which contains the ID of the resource, the IDs of the both parties, the granted rights, a validity period and Bob's signature. Alice saves the capability to her capability list. She can now encapsulate the capability into a service request (e.g. an HTTP GET request) and send it to the access decision service [11].
The majority of the described IAM frameworks addresses the issue that IoT devices tend to be resource constraint by holding only a small share of data and logic on the devices. While Salman et al. [26] address the device heterogeneity problem with an additional controller layer, the other frameworks do not explicitly explain how devices with differing hardware and software can be authenticated or authorized. All IAM frameworks for IoT described in this Section commonly leverage central trusted entities to perform IAM operations. Centralization generally implies reliance on a single-point-of-failure. This means that any vulnerability could enable compromisation of a large stake of a system and its data. A centralized approach does not support end-to-end security. Users need a trust relationship because data security and privacy cannot be reviewed transparently. Ouaddah et al. [21] further state that centralized IAM may become too expensive in large networks in the long term.

3 IAM Challenges in Context of IoT

IoT implies several constraints for enterprise IAM. A number of recently published research indicates the different requirements. After analyzing the content, we were able to derive the following generalized challenges based on the body of literature as well as on our practical experience in IAM. However, note that our goal is not to present an exhaustive list but rather discovering the most important challenges of IAM in the IoT context:

- Physical design constraints (e.g. mobile devices with low power)
- Need for comprehensive and secure IAM mechanisms
- Variability of identities (e.g. interoperability of heterogeneous devices)
- Network scalability

IoT devices (e.g. an RFID tag on clothes) often do not have a high computational capacity and are low powered. Therefore different constraints based on their **physical design** arise. Such devices are not able to execute highly demanding cryptographic operations. This must be especially considered when it comes to authentication and access control within an enterprise IAM. Additionally, IoT devices need to be replaced more often because of their design constraints. Thus, compared to traditional scenarios, the IAM lifecycle has to be executed far more often when IoT applications are involved [1].
This leads directly to the second challenge, the **need for comprehensive and secure IAM mechanisms**. As identities within an IoT context are very large in number it has to be ensured that each device has a managed identity within a supervised IAM platform, information about the identity of all other devices, and the possibility to verify them. Then and only then one can provide a comprehensive view on all identities within an enterprise IoT as well as a secure way for collaboration [33].
One of the further challenges restricting this is the **variability of identities**. Within traditional IAM, identities were mainly humans (e.g. employees or customers). However, within IoT most of the identities will be of non-human origin. They are highly heterogeneous as they have different attributes which need to be managed correctly. An employee holds, for example, attributes regarding his department while an IoT device is assigned an attribute *software version* tracking the current status of its software. To ensure **interoperability**, the IAM environment must be able to manage those attributes, data sources, and policies from different sources [25].
Finally there must be a **scalable mechanism** to manage device identities, authentication, and authorization to enable trusted interactions between devices [25]. An IAM platform has to ensure full operability regardless the number of managed identities and the accompanying requirements for storage and network consumption. However, current approaches are not able to fulfill this as the number of identities within a network is far beyond present numbers [33].

4 Blockchain-based IAM in an Enterprise Context

Having outlined major challenges for IAM when it comes to regulating IoT, in this Chapter we take a closer look at the blockchain technology (Section 4.1) and how it can potentially be applied to address these obstacles (Section 4.2).

4.1 Blockchain Technology: Beyond Cryptocurrencies

Blockchain technology was first introduced as enabling technology for the Bitcoin cryptocurrency. Bitcoin implements a blockchain network, i.e. a decentralized set of nodes which all hold a valid copy of the blockchain. The network must establish consensus on the chronology of transactions to establish an authoritative, final transaction log on all nodes [6]. In so-called public blockchains such as Bitcoin, access to the network is not restricted. Thus, anybody can join and participate.

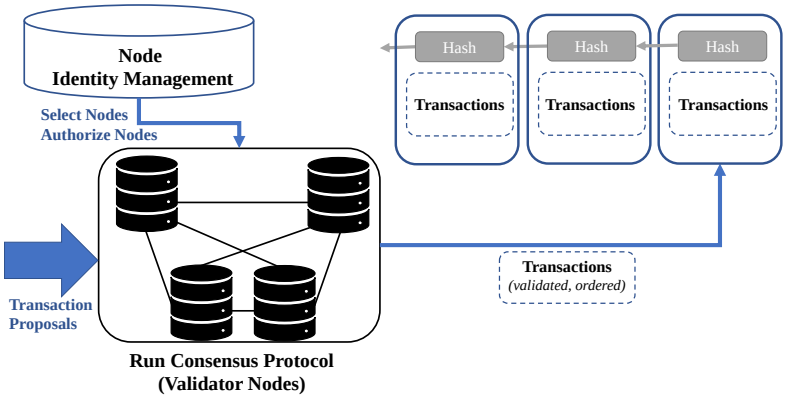

Fig. 3. Simplified Validation Process for a Private Blockchain

To prevent public blockchains from being vulnerable to sibyl attacks [8], computationally difficult consensus mechanisms such as proof of work (PoW) in Bitcoin are applied. However, temporary collisions can emerge in public blockchains due to network delays which require the application of conflict resolution rules [31]. In private blockchain networks, participants are known and whitelisted. Such a system is also referred to as private or permissioned blockchain. The nodes which establish a private blockchain must be initially authorized by a trusted authority [29]. This process can be referred to as **node identity management** [31]. Up to now, several blockchain frameworks have been developed for different purposes and with different design and functional properties. The Hyperledger project is one well-known example which aims at the establishment of an open standard for blockchain-based enterprise business transactions. From a security point of view, the Hyperledger architecture intends node authentication via a certificate authority which distributes enrollment certificates to the nodes. Transactions are encrypted using a symmetric key per blockchain which all peers in the network hold. For future versions, more fine-grained encryption for transactions is planned [4].

Due to node identity management, private blockchains can rely on computationally inexpensive, voting-based consensus mechanisms, thus enabling processing of tens of thousands of transactions per second. One class of consensus mechanisms for private blockchains which is currently employed is based on the **Byzantine Fault-Tolerant (BFT) Protocol** [16]. BFT consensus mechanisms offer **consensus finality** which means that all correctly working nodes will process blocks within their copy of blockchain in the same way (e.g. by applying the same rules and policies). This condition prevents the emergence of collisions [31]. The fault assumption underlying BFT consensus requires that among n validating nodes, the number of nodes behaving arbitrarily does not exceed 33% [4]. Figure 3 illustrates a simplified transaction validation process in a private

blockchain. An entity which wants to submit data to the blockchain encapsulates the request into a transaction and proposes it to the validating nodes. The validating nodes enforce a given set of contract rules by the replicated execution of **smart contracts**. Smart contracts are self-executing scripts which can enforce the properties of an arbitrary digital contract. A smart contract can be triggered by issuing a transaction to its unique address on the blockchain [6]. The consensus protocol ensures that the transactions applied on each node do not diverge. By means of state-machine replication [28], consistent replication of a smart contract in a decentralized network can be realized [31]. In case of Hyperledger Fabric a BFT-based consensus mechanism is applied [4]. To prevent diverging states between the nodes, smart contracts need to be deterministic. In case of Hyperledger Fabric, smart contracts can be installed to each node in a blockchain network by issuing a so-called deploy transaction. Figure 3 further indicates that node identity management is required to initially select validating nodes. Dynamically changing sets of validating nodes are planned for future versions of Hyperledger Fabric. Non-Validating Nodes are supported which receive transactions and forward them to the validating nodes [4].
Vukolic [31] states that a BFT network always maintains its correct state and consensus finality despite arbitrarily long asynchrony. According to Fischer et al. [9], faulty nodes can lead to a state in which consensus is never reached when a network is entirely asynchronous. Integrity of the blockchain would still be maintained in this scenario. However, the system would be prevented from making further consensus decisions. Thus, availability of the service might be affected.
In the following section, recently published frameworks and approaches which apply the blockchain technology to IAM within an IoT environment will be discussed under consideration of the challenges described in Section 3.

4.2 Blockchain-based Enterprise IAM for IoT

Recent research literature outlines promising approaches and ideas regarding the use of blockchain technology to enhance the security of specific IAM functions for application in an IoT scenario. Even though the ideas presented in the following are to some extent designed for public blockchain use-cases, we find that the adaption to the organizational context could be beneficial regarding the challenges discussed in Section 3. Evaluation of advantages and disadvantages and description of a practical enterprise use-case will take place later on. **Identity Management.** Zhu et al. [33] propose a blockchain-based identity framework for IoT (BIFIT) for smart home environments. The framework enables the management of IoT devices by their respective device owner. Owner identities are held on a blockchain and managed by transactions. Owners randomly create the key pairs used to generate identifiers and credentials from the same seed used for their own identity. Device identities further contain the owner's signature as an attribute. This approach can be applied to all kind of IoT devices, thus ensuring **interoperability**. The digital identities of owners are created by issuing a transaction to a blockchain which contains an identifier hash value, key pairs, the identity signature, and a storage pointer. The owner identity is stored in the

blockchain in a **tamper-proof** way and can be used for validation. Update or revocation transactions can be performed to revoke or update an owner's identity. Lookup transactions for information retrieval can be used to enable reliable authentication. IoT devices only need to store the block header of the identity chain to authenticate other devices. By this means, resource usage is limited to enhance **suitability for resource-constraint devices**. The authors predict a growth rate of blockchain which is far lower than the rate of the Bitcoin network because transactions can only be issued by owners and not by all nodes. This supports **scalability in terms of network and storage consumption** [33].

Access Control. Maesa et al. [17] leverage blockchain to store representations of access rights to a specific resource in a **tamper-proof** way and to manage those rights via blockchain transactions. Access rules are employed by attribute based access control (ABAC) [14] policies. Policies consist of conditions defining a set of allowed values for attributes and specify the actions which subjects are entitled to perform on the addressed resource. Attributes can be related to the subject demanding access, the resource, or the environment. Policies are initially defined by the resource's owner who issues a policy creation transaction to the blockchain. Resource owners can update and revoke policies by issuing update or revocation transactions to the blockchain respectively. Resource owners can change their policies over time. All changes such as policy updates and right transfers are timestamped and logged to the blockchain in a traceable way. Resource owners can issue right transfer transactions which are linked to a particular subject. When receiving a subject's request for access to a resource, a policy enforcement point authenticates the subject by its id and a challenge and queries the blockchain for transactions holding relevant policy data. It then builds a standard XACML policy [10] which is transferred to a policy decision point where it is evaluated against the subject's attributes. Maesa et al. state that putting policy evaluation and execution in a smart contract will be subject to their future research work [17]. The IoT context is not directly addressed in the work of Maesa et al. However, the framework only requires subjects to hold an ID and to sign a challenge which may contribute to **portability to resource-constraint devices**.
Shafagh et al. [29] leverage the blockchain technology to manage ownerships and sharings of data streams provided by IoT devices. Owners can share data streams by issuing a new transaction to the blockchain which holds the identifier of the data stream and the service's public key. The potential impact of a node acting maliciously is limited because each node only holds a small encrypted piece of a data stream. A user who wants to revoke access rights to a data stream changes the encryption key and shares it with all authorized services except the one to be revoked. Additionally, the owner issues a new transaction which replaces previous permissions. This also facilitates monitoring of access management activity. The blockchain does not hold these chunks but only their hash pointer to the previous chunk. It contains a hash pointer of each chunk to ensure **tamper-proof storage**, i.e. integrity. The authors propose a decentralized and encrypted

storage layer to further ensure confidentiality of the stored stream data. Besides **security** this also supports **scalability in terms of storage consumption** because only a hash pointer needs to be appended to the immutable blockchain [29].

Storage and Monitoring. Polyzos & Fotiou [23] emphasize that the tamper-proof storage property is beneficious for the development of robust monitoring mechanisms. Users cannot deny having approved a transaction because the authenticity of blockchain is verified by a network of nodes. An attacker would have to forge a digital signature and gain control over a larger share of nodes in the network to alter information held within a blockchain [29]. Thus, only valid transactions can be kept within a blockchain which ensures **non-repudiation** of the logged information [23].
Azaria et al. [2] demonstrate that a blockchain which incorporates smart contracts can provide powerful backup and monitoring functionality. Due to the decentralized nature of storage, a complete log of the issued transactions will remain in the blockchain, no matter whether a user leaves or rejoins the network over an arbitrary period of time. Access to the respective log only requires the download of the latest version of the blockchain. The blockchain log is maintained as long as there are nodes in the network [2].
Maesa et al. [17] state that the persistent, immutable storage of data in a blockchain requires the definition of a protocol which minimizes the amount of data necessary per transaction to enable **scalability in terms of storage space requirements** in the entire network. An easy solution would be the storage of a record containing a link to the data stored in an external database together with a cryptographic hash of the data to store. This approach would still hold the data in a tamper-proof way while needing significantly less storage. However, external storage would not achieve the benefits of decentralized storage in the blockchain in terms of other protection goals such as availability. Thus, Maesa et al. [17] propose an approach where the entire information is stored in the blockchain in an encoded format which aims at rewriting data fields to a representation with constant size. The easiest example would be rewriting the operators AND, OR to the numerical representations 0, 1. To achieve a mapping of attribute names and values in representations of a length of e.g. 1 byte, publicly available conventions must be maintained [17]. This approach might also be applicable to all kinds of structured data which is of interest for monitoring, e.g. access logs to a resource involving a timestamp, a user id, and a target resource. By this means, more scalable blockchain-based monitoring approaches in terms of storage space requirements could be achieved [17].

At this point, we outlined an abstract view of how the blockchain can be applied to the different IAM functions. Figure 4 illustrates a simplified model of the potential interaction of the different IAM functions and the blockchain. It is based on the blockchain implementation proposed by the research work presented in this section. It shows at which position of a generic IAM process adapting the

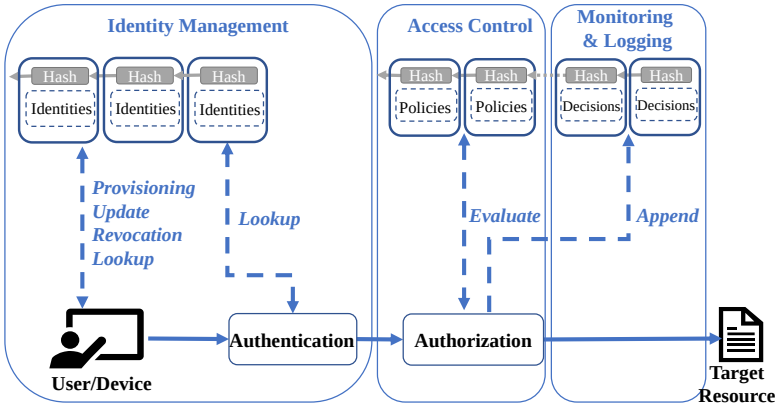

Fig. 4. Blockchain and the IAM functions

blockchain can be applied usefully and provides an example per IAM operation. Figure 4 illustrates that blockchain technology can be applied to all three basic IAM operations, namely identity management, access control, and monitoring. In the following Section, a use case will focus on the potential applicability of blockchain to IAM in the enterprise domain.

5 Application Scenario for an Enterprise IAM

In this Chapter, a comprehensive use case for the application of the blockchain technology to IAM and enterprise IoT will be described. It addresses the challenges described in Section 3 and applies some of the ideas presented in Section 4. As the scope of enterprise IAM is usually restricted to corporate systems and users, a private blockchain is assumed in the following. Figure 5 illustrates the use case including all three main components of modern enterprise IAM mentioned within Section 1. Each color highlights a specific component.

Let's assume a manufacturing company where employees leverage the support of smart devices. The devices automatically trigger actions based on the analysis of sensor data provided by the machines working in an assembly line. Entitled employees or devices should be able to access sensor data which is aggregated in a sensor data storage as illustrated in Figure 5. A private blockchain is applied to support all relevant IAM functions as noted above. Thus, each employee and each smart device needs an identity containing different kind of master data such as personal information and references to a department, team, location, etc. The hash of the identity is used as a unique identifier. The **identity management** process which is illustrated by green arrows in Figure 5 is initiated by entering the identity information of a new employee or device into the system. The encrypted identity information is stored in a central *identity store*. Besides that,

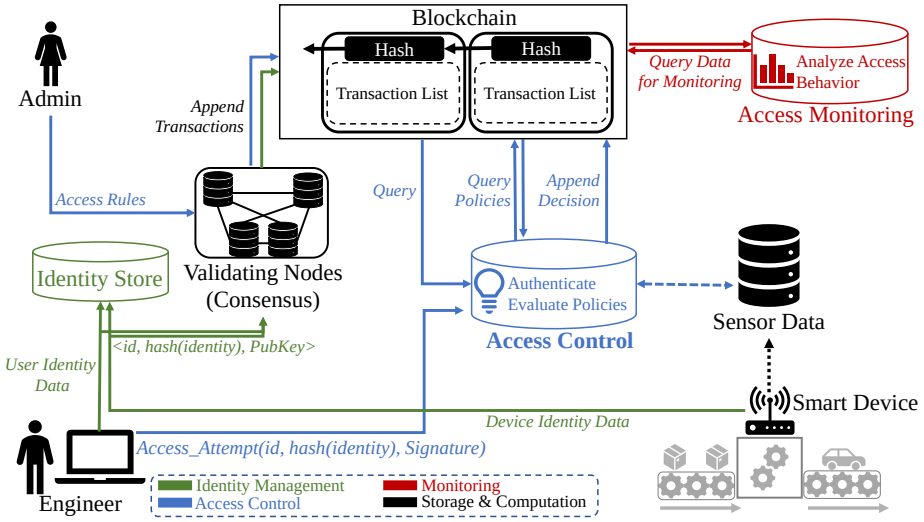

Fig. 5. Potential Blockchain-based Enterprise IoT Scenario

a key pair is created and together with the hash of the identity encapsulated into a blockchain transaction which is signed with the private key. In the case of device identities, the firmware and configuration running on the device can be hashed and stored on the blockchain to make potential unauthorized changes to the device traceable. Devices will be rejected from access when unauthorized alterations of the firmware or configuration are recognized (as e.g. proposed by [15].

After issuing the transaction, each node in the private blockchain network runs a *consensus protocol*, validates the transaction, and *appends* it to the blockchain. It should be noted that this is a simplified example and does not utilize a specific protocol. Update and revocation transactions are triggered when identity updates or the deletion of an identity by the employee or an authorized administrator take place. If now, for example, a device initiates communication e.g. via smartphone of an employee to deliver sensor data, the employee can A) identify the device securely by validating its signature using the public key stored in the blockchain. B) The employee can ensure that the device was not manipulated by comparing the hash value stored in the blockchain with the hash value of the actual data delivered with the request.

To protect critical resources from unauthorized access, suitable access control needs to be applied. In Figure 5, **access control** operations and entities are illustrated by the blue-colored arrows and entities. Our access control scheme is based on the commonly used ABAC defined by Hu et al. [14]. The scenario starts with an administrator who creates new access control policies and issues them as transactions to the blockchain (*access rules*). For example, employees

and devices could hold an attribute *"emergency operations"*. A policy could then enforce that all users or devices which hold this specific attribute are able to access temperature sensor data of engines. However, additional attributes are necessary to read or write more critical information such as prototype blueprints which can be sent to the construction plant.
In the scenario illustrated in Figure 5 the access control component receives the access request of a user containing the user's attributes (*access_attempt*) and queries the relevant policies from the blockchain (*query policies*). It then assesses the policy rules against the attributes and makes a decision (*authenticate and evaluate policies, append decision*). In a next step, access decision logic could be implemented directly on the blockchain using smart contracts.

Under the aspect of **monitoring** (in color red) all nodes appended to the blockchain can be monitored as the blockchain itself can be regarded as a log storage. Any user action gets appended to the blockchain including the identity's unique identifier as well as additional information regarding its action. This could be the respective entitlement that was evaluated or any further attributes available (e.g. IP address or timestamp). Malicious activities can then be detected by comparing the identity's behavior with the historic usage pattern which remains within the blockchain and can be retrieved by traversing the blockchain and searching for all logs with the identity's unique identifier (*query data for monitoring*). An advantage compared to traditional logging mechanisms is the decentralized data storage within the blockchain. A malicious attacker cannot easily manipulate the log collection (e.g. by deleting logs after an attack) by targeting only one log server. Because the logs are stored on many different devices via blockchain technology an attacker would need to maintain control over a specific amount of devices (based on BFT this would be more than 33% as discussed in Section 4.1).

6 Discussion

As described in Section 1, the secure and reliable implementation of IAM functions is a precondition to maintain enterprise security. Beyond comprehensive security, IoT requires IAM to deal with additional challenges such as device constraints, heterogeneity, and scalability. However, traditional centralized approaches imply a single-point-of-failure which poses the threat of compromise or failure of highly security-relevant IAM functions such as identity provisioning, permission assignment, access control, or monitoring of user or device behavior in case of e.g. unauthorized data tampering.
All operations which are performed by issuing transactions are immutably logged to the blockchain, thus enabling a transparent, tamper-proof log of all security-relevant operations. However, Section 4.1 shows that **public blockchains** lack scalability in terms of large numbers of clients and transactions due to very low transaction processing rates. This reasons the usage of computationally intensive consensus mechanisms. Beyond that, Section 4.1 has shown that **private**

blockchains offer proper scalability in terms of large numbers of clients and transactions. This makes private blockchains more suitable for common enterprise application scenarios than public blockchains.
The results of the frameworks within Section 4.2 further indicate that scalability in terms of space requirements can be managed. The same applies for device interoperability and suitability for resource constraint devices. However, it is worth noting that the latter two challenges depend on the concrete implementation and cannot be solved by the mere deployment of a blockchain. Nonetheless, the research work in Section 4.2 addresses those challenges successfully as the design of their frameworks for instance does not require the storage of the entire blockchain on devices, or takes measures to keep blockchain growth on a moderate level.
Private blockchains imply several limitations which may affect potential IAM applications for IoT in enterprises. As stated in Section 4.1, the implementation of a private blockchain currently requires a trusted entity at least for initial node identity management. This could as well constitute a single-point-of-failure if a centralized entity is elected for this task. However, according to Rodrigues et al. [24], dynamic reconfiguration of system memberships in BFT-based systems are generally possible after initial election. While private blockchains provide significantly better scalability in terms of large numbers of clients, BFT protocols which are currently employed in several private blockchain architectures are generally considered as not scalable towards increasing numbers of nodes due to a large network latency [18]. This is due to the large number of network interactions required between nodes. However, according to Vukolic [32], node scalability has not been tested intensively yet beyond a network size of 20 nodes. From a security point of view, a too small set of nodes could relativize the increase in security as it might become easier for an attacker to reach the critical number of compromised nodes. As this is a theoretical consideration, further research could examine whether the number of nodes has a practical implication on a system's vulnerability against Sybil attacks which might, for instance, be performed by malicious insiders with sufficient permissions. Even though private blockchains surpass public blockchains in terms of performance and scalability, further research will benchmark whether the performance of private implementations meets the requirements of time-critical use-cases.

7 Conclusion

The rise of IoT poses new challenges for IAM in enterprises. The purpose of this paper is to assess how blockchain-based IAM can deal with theses obstacles, and to demonstrate this in a practical use-case. IoT implies an increased demand for secure and comprehensive IAM operations, interoperability between heterogeneous devices and support for resource constraint devices as well as scalability in terms of network and storage consumption. Private blockchains can contribute several important properties for the enterprise context such as secure tamper-proof storage of IAM data without reliance on a single-point-of-failure. Beyond

that, private blockchains provide much better scalability towards large numbers of clients and transactions than public blockchains. Our research indicates that scalability in terms of space requirements, interoperability, and support for resource constraint devices in IAM scenarios are manageable. However, private blockchains might still require a central trusted entity for (initial) node identity management and lack scalability regarding larger networks of nodes. The use case presented in Section 5 demonstrates that blockchain can pose a reliable data storage for all major IAM operations and can form the basis for the design of comprehensive IAM architecture in common enterprise IoT scenarios.

Acknowledgment. This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de).

References

1. Adireddy, A., Gottapu, U., Aravamudhan, A.P.: Usercentric federation of access to internet-of-things(iot) devices: A valet key for iot devices. In: 2016 International Conference on Circuits, Controls, Communications and Computing (I4C). pp. 1–7 (Oct 2016). https://doi.org/10.1109/CIMCA.2016.8053280
2. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: Using blockchain for medical data access and permission management. In: Open and Big Data (OBD), International Conference on. pp. 25–30. IEEE (2016)
3. Babar, S., Mahalle, P., Stango, A., Prasad, N., Prasad, R.: Proposed security model and threat taxonomy for the internet of things (iot). Recent Trends in Network Security and Applications pp. 420–429 (2010)
4. Cachin, C.: Architecture of the hyperledger blockchain fabric. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers (2016)
5. Chen, J., Liu, Y., Chai, Y.: An identity management framework for internet of things. In: e-Business Engineering (ICEBE), 2015 IEEE 12th International Conference on. pp. 360–364. IEEE (2015)
6. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016)
7. Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V.: Blockchain technology: Beyond bitcoin. Applied Innovation **2**, 6–10 (2016)
8. Douceur, J.R.: The sybil attack. In: International Workshop on Peer-to-Peer Systems. pp. 251–260. Springer (2002)
9. Fischer, M.J., Lynch, N.A., Paterson, M.S.: Impossibility of distributed consensus with one faulty process. Journal of the ACM (JACM) **32**(2), 374–382 (1985)
10. Godik, S., Moses, T.: Oasis extensible access control markup language (xacml). OASIS Committee Secification cs-xacml-specification-1.0 (2002)
11. Gusmeroli, S., Piccione, S., Rotondi, D.: Iot access control issues: a capability based approach. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. pp. 787–792. IEEE (2012)
12. Haller, S., Karnouskos, S., Schroth, C.: The internet of things in an enterprise context. In: Future Internet Symposium. pp. 14–28. Springer (2008)
13. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly **28**(1), 75–105 (2004)

14. Hu, V.C., Ferraiolo, D.F., Kuhn, D.R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K.: Guide to attribute based access control (abac) definition and considerations. NIST Special Publication pp. 800–162 (2014)
15. Kshetri, N.: Can blockchain strengthen the internet of things? IT Professional **19**(4), 68–72 (2017)
16. Lamport, L., Shostak, R., Pease, M.: The byzantine generals problem. ACM Transactions on Programming Languages and Systems (TOPLAS) **4**(3), 382–401 (1982)
17. Maesa, D.D.F., Mori, P., Ricci, L.: Blockchain based access control. In: IFIP International Conference on Distributed Applications and Interoperable Systems. pp. 206–220. Springer (2017)
18. Mickens, J.: The saddest moment. Login Usenix Mag **39**(3) (2014)
19. Moinet, A., Darties, B., Baril, J.L.: Blockchain based trust & authentication for decentralized sensor networks. arXiv preprint arXiv:1706.01730 (2017)
20. Osmanoglu, E.: Identity and Access Management: Business Performance Through Connected Intelligence. Newnes (2013)
21. Ouaddah, A., Mousannif, H., Elkalam, A.A., Ouahman, A.A.: Access control in the internet of things: Big challenges and new opportunities. Computer Networks **112**, 237–262 (2017)
22. Outchakoucht, A., Hamza, E.S., Leroy, J.P.: Dynamic access control policy based on blockchain and machine learning for the internet of things. International Journal Of Advanced Computer Science And Applications **8**(7), 417–424 (2017)
23. Polyzos, G.C., Fotiou, N.: Blockchain-assisted information distribution for the internet of things. In: 2017 IEEE International Conference on Information Reuse and Integration (IRI). pp. 75–78. IEEE (2017)
24. Rodrigues, R., Liskov, B., Chen, K., Liskov, M., Schultz, D.: Automatic reconfiguration for large-scale reliable storage systems. IEEE Transactions on Dependable and Secure Computing **9**(2), 145–158 (2012)
25. Roman, R., Zhou, J., Lopez, J.: On the features and challenges of security and privacy in distributed internet of things. Computer Networks **57**(10), 2266–2279 (2013)
26. Salman, O., Abdallah, S., Elhajj, I.H., Chehab, A., Kayssi, A.: Identity-based authentication scheme for the internet of things. In: Computers and Communication (ISCC), 2016 IEEE Symposium on. pp. 1109–1111. IEEE (2016)
27. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. Computer **29**(2), 38–47 (1996)
28. Schneider, F.B.: Implementing fault-tolerant services using the state machine approach: A tutorial. ACM Computing Surveys (CSUR) **22**(4), 299–319 (1990)
29. Shafagh, H., Hithnawi, A., Duquennoy, S.: Towards blockchain-based auditable storage and sharing of iot data. arXiv preprint arXiv:1705.08230 (2017)
30. Trnka, M., Cerny, T.: Identity management of devices in internet of things environment. In: IT Convergence and Security (ICITCS), 2016 6th International Conference on. pp. 1–4. IEEE (2016)
31. Vukolić, M.: The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In: International Workshop on Open Problems in Network Security. pp. 112–125. Springer (2015)
32. Vukolić, M.: Rethinking permissioned blockchains. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. pp. 3–7. ACM (2017)
33. Zhu, X., Badr, Y., Pacheco, J., Hariri, S.: Autonomic identity framework for the internet of things. In: Cloud and Autonomic Computing (ICCAC), 2017 International Conference on. pp. 69–79. IEEE (2017)