THE JBBA

PEER REVIEWED RESEARCH

OPEN ACCESS
ISSN Online: 2516-3957
ISSN Print: 2516-3949
https://doi.org/10.31585/jbba-4-1-(5)2021

Identity of Things: Applying concepts from Self Sovereign Identity to IoT devices

[1]Tim Weingärtner, [2]Oskar Camenzind
[1]Blockchain Lab, Lucerne University of Applied Sciences, Switzerland
[2]Building Technologies Division, Siemens Schweiz AG, Switzerland

Correspondence: Tim.Weingaertner@hslu.ch

Received: 31 December 2020 **Accepted:** 20 February 2021 **Published:** 24 February 2021

Abstract

Identity is a crucial property of Internet of Things (IoT) devices. Due to rapid growth and high numbers of similar devices, reliable identification of those devices is a problem. The origin and history of an IoT device is especially important in security-relevant environments.

Our research addresses this issue by proposing an approach based on blockchain and decentralised identifiers (DID). It is inspired by the concepts of self-sovereign identity (SSI) and bootstrapping of remote secure key infrastructures (BRSKI). Devices are equipped by the manufacturer with an identity stored in a trusted execution environment (TEE) and secured by a blockchain. This identity can be used to trace back the origin of the device. During the bootstrapping process on the customer side, the identity registration of the device is updated in the blockchain. This process is performed by a so-called registrar. Smart contracts prevent unsolicited transfer of ownership and track the history of the device. Besides proof of origin and device security our concept can be used for device inventory and firmware upgrade.

A prototype implementation was realised to validate the concept. All six use cases have been implemented and tested using an Ethereum blockchain infrastructure. JSON Web Tokens (JWT) have been used as signed artefacts to transfer information between the stakeholders. This enables an asynchronous communication needed for example in an environment with no direct internet access. Such an infrastructure can be provided by an independent association and can be used by all manufacturers. Depending on the environment a registration of devices can be optional or mandatory.

Keywords: *IoT, identity, blockchain, self-sovereign identity, decentralised identifier (DID), json web token (JWT)*

JEL Classifications:

1. Introduction

The rapidly growing number of devices used for the Internet of Things (IoT) is raising concerns about the origin and history of these devices. Security issues regarding IoT devices lead to new concepts about bootstrapping and administration. Identity becomes a crucial property of IoT devices. So far there are primarily proprietary solutions. In a multi-provider environment those kinds of approaches have major disadvantages since the customer himself is responsible for administration.

In this paper we propose a new approach applying concepts from self-sovereign identity to IoT devices ensuring their identity and history. Derived from [1] we call our approach a "Manufacturer Authorized Signing Authority Blockchain Infrastructure" (MASA-BI). As the name implies, the system is based on the blockchain technology to ensure immutability, autonomy and unified interfaces.

First, we will give a short introduction into the topic of identity and self-sovereign identity summarizing the major concepts used here. The related work shows the already existing approaches and illustrates the previous knowledge our approach is based on. Our approach consists of six use cases (UC1–UC6) which are arranged around two main application areas. The analysis of advantages and disadvantages as well as a final conclusion completes the paper.

2. Identity

When speaking of identity, the first thing that comes to mind is the identity of a person. *Webster*[1] defines identity as "the distinguishing character or personality of an individual". Beside the psychological aspects of identity, we use it to distinguish persons from each other. The identity check uses attributes of an entity to verify if a person is the one, he or she claims to be.

[1] https://www.merriam-webster.com/dictionary/identity

Those attributes can be physical or non-physical. Physical or physiological attributes which define an identity are fingerprints, face, iris structure, voice, DNA, smell, speech, location as well as possession or access to physical objects like identity card, mobile device, notes, etc. Non-physical attributes which define our identity mainly depend on our brain like knowledge, abilities, memories, experiences, relationships, feelings, wishes, behavior or secrets. For an identity check we compare those attributes with previously stored data. Most of the time we use a combination of different attributes. At the airport, the identity card a person posseses is checked against his appearance. In addition, biometrical data like face lineaments are compared. When a password is requested the knowledge of an individual is checked, sometimes in combination with a message to the mobile phone which should be in possession and access of this person. The attributes can be classified by their difficulty to copy, steal, or guess them.

The identity of a "thing" has some similarities to those of a person even though a thing can be copied. For example, each specimen of a certain sensor is identical if we do not get on an atomic level. We can give them an identity by adding individual attributes like a serial number. If a sensor has a memory chip, its "experiences" can make it different to a similar device.

But why do things need an identity? If we want to move into the direction of a digital twin - the digital copy of a physical object - identity is crucial [2]. Each data point which is detected in the real world has to be assigned to the corresponding position of the digital twin. Errors or fraud have to be excluded. Otherwise, the digital twin is just an anonymous copy.

Securing this identity is a big challenge today and there is a lot of research going on in this area [3]. Since all digital data can be copied easily one has to take steps to avoid this and protect the identity of a device. Most common, secured elements are used, that make it hard to impossible to access those data. To avoid the copying of data at the interface level the data has to be signed by the device. It has to be kept in mind that the needed processing power for the cryptographic calculations of the signing process has to be provided by the device.

2.1. Self-sovereign identity

Self-Sovereign Identity (SSI) allows a person to create her own identity and get a verification or proof by a trusted third party such as the government. Although SSI is independent of blockchain technology it is often used together. Blockchain has seen a rise in importance as a technology to store data in an immutable way there therefore to guarantee and confirm identity. Systems like uPort[2] or Sovrin[3] together with Hyperledger Indy[4] are just some examples of existing solutions. Since no personal data is stored on the blockchain, the compliance to GDPR (General Data Protection Regulation) is assumed [4]. There is still some doubt about it and clear guidelines from the regulators are demanded [5].

With the concept of self-sovereign identity using Decentralized Identifiers (DIDs) [6] it is possible to store identities and verifiable claims on the blockchain. The DID is a globally unique identifier which does not need an explanation since its DID scheme links to a specific method explaining how the DID is resolved and links to a DID document describing all details. The DID document is fully self-describing and contains information about the entity the DID is about. This includes cryptographic information or service endpoints. For GDPR compliance reasons it is important that neither DID nor DID document contain person-related information.

A DID looks like:

did:ethr:0xe34eac30c498d9e26865f64fcaa57dbb935b0d7a
and consists of three parts separated by a colon:

1. String "did" for the URL scheme
2. DID method[5]
3. Specific identifier

While the DID represents the identity of the entity, additional verifiable claims describe qualities or properties of the entity [7]. Those claims have to be issued by a trusted party which itself is represented by an identity (DID). Verifiable claims can be stored on a blockchain to ensure immutability and independence from the availability of the issuer. The uPort in [8] shows an example of such an ecosystem. While claims are stored for example in a smart contract on a blockchain, JSON Web Tokens (JWT) can be used to transfer and interchange verifiable claims off-chain [9][10].

JWT consists of three parts separated by dots [11]:

1. Header, with information about the signing algorithm
2. Payload, containing the claim
3. Signature, which is the signed header and payload

To reduce the size, the header and the payload are Base64Url encoded.

The claim itself contains information about the issuer and the date of issuing, the subject or entity the claim is about, the audience the claim is intended for and optionally an expiration time. Further optional fields are possible. Examples and libraries for JWT can be found on jwt.io[6].

[2] https://www.uport.me/
[3] https://sovrin.org/
[4] https://www.hyperledger.org/projects/hyperledger-indy
[5] https://w3c-ccg.github.io/did-method-registry/
[6] https://jwt.io/

3. Related work

Self-sovereign identity of persons is discussed in several research papers [12][13][14][15]. Some of them cite the ten key properties of self-sovereign identity from C. Allen [16]:

1. Existence of the entity in the real world
2. Control from the entity over its identity
3. Access to the own data
4. Transparency about the systems and algorithms used
5. Persistence and long-liveness of the identity
6. Portability of the identity to guarantee independence of systems
7. Interoperability of the identity through open standards
8. Consent of the entity to share or use their identity
9. Minimalisation of data that is disclosed through a claim
10. Protection of the entities' rights

Al-Bassam describes in his paper [17] a smart contract-based identity system where each entity is represented by an Ethereum address. His SCPKI system focusses on persons or organisations as entities which control their identity over the private key to their Ethereum address. A claim or proof is reduced to a Boolean value in the attributes of an identity.

Self-sovereign identity is seen by Der et al. [18] as one of the essential enablers for a digital revolution. In the outlook of their paper the usage of self-sovereign identity for things is mentioned as future research area. Conceptional questions like "How can a non-human entity recognize and characterize its own identity", are raised.

A first overview about self-sovereign identity for Industrial Internet of Things (IIoT) is presented by Bartolomeu et al. [19]. Their paper provides a review of several use-cases and challenges Self-Sovereign Identity face in the context of IIoT. One application mentioned is the authentication of devices. It is mentioned that most solutions rely on a centralised instance and blockchain-based SSI is one possibility to overcome this drawback.

4. Giving a device an identity during manufacturing

As described above, a device has to "receive" an identity to act as a unique digital twin. Since this identity is not linked to physical uniqueness it is an artificial act. Therefore, this is security wise a critical moment and should be done during manufacturing and in a secured environment. There are several possibilities to include a secured environment on a chip to store this identity in a save way. Trusted Execution Environments (TEE) represent one solution for it. Shepherd et al. [20] give an overview of actual technologies. Companies like Intel, LEGIC[7] or Riddle & Code[8] provide products to store private keys in a secure element on a chip. In our approach we leave this intentionally to the manufacturer.

We propose a system containing a smart contract DIDManufacturerInventory which manages the identities of IoT devices. Our prototype is based on the Ethereum blockchain and its signature system since this infrastructure offers the broadest development environment. The implementation can be easily ported to another blockchain environment that offers similar features. While each device holds its address and the access to it as private key the proposed smart contract acts as proof of origin of the device. In our proposal the device manufacturer generates a private key and an Ethereum address (derived from the public key) according to the Ethereum address requirements [21] and stores this in a secure area on the device. Either way, once this identity is created on the device as required, or if the device is used in a secure environment, we assume that the device eventually contains its private key, which cannot be accessed from outside the device. Since the private key grants access to the blockchain the device now has a) access to its address on the Ethereum blockchain and b) can sign messages with its private key. The access to the blockchain is not required for our approach since we want to avoid high resource consumption.

In a first step the manufacturer generates his own Ethereum address and registers to the DIDManufacturerInventory once. This is the first use case which has been implemented in our prototype (UseCase1 = UC1). We intentionally decided not to require proofs for the identity of manufacturers to reduce the hurdle of participating in such a system. At a later stage this can be introduced easily. With his account address the manufacturer can register as many devices as wanted. Each device receives its own Ethereum address as describe above. The device registration process is the second use case (UC2). It stores the public key of the device in the smart contract. For data privacy reasons a manufacturer can possess more than one address on the blockchain (UC1). Besides this, no identifying data is stored on the blockchain. During UC2 trusted public keys of MASA nodes have to be stored on the device. We will see later on the purpose of this measure.

5. Bootstrapping a device in a new environment

The second part of our proposal is related to the registration in the client environment. Once the device is shipped and installed at the premises of the customer the bootstrapping process begins (see Figure 1). The registrar has the role of an onsite registration authority. Usually, one registrar per site is foreseen and a 1:1 relation between device and registrar will hold for most cases. Nevertheless, it is also possible to use several registrars which we will see in UC6.

[7] https://www.legic.com/

[8] https://www.riddleandcode.com/

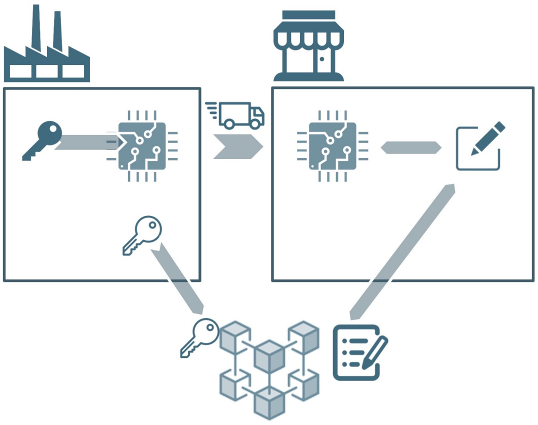


Figure 1: Overall process from manufacturing to registration in the client environment.

UC3 represents the initial registration use case of the registrar in the smart contract DIDInventory. This process is very similar to the registration of a manufacturer (see UC1) and is as well self-sovereign. Due to design reasons, we separated the identity distribution during manufacturing (UC1 and UC2) from the bootstrapping at client environment (UC3–UC6) by two separate smart contracts. It can be decided at a later stage if one blockchain for both environments should be used or if they are to be kept separately. We discuss the advantages and disadvantages in section 7.

After the registrar is authorised as such, the bootstrapping of the device can start (UC4). This process is derived from BRSKI [1]. It can be initiated by the device looking for a registrar through first boot up, via a manual action like pressing of a button, or by accessing the device with an initial call. The bootstrapping process UC4 contains 10 steps (see Figure 2):

1. **Device** informs the surrounding that it is active or is initially called.
2. **Registrar** sends its identity (public Ethereum address) to the device.
3. **Device** includes the registrar's identity in a JSON Web Token JWT1 and signs this token with its private key and sends it to the registrar.
4. **Registrar** includes JWT1 into a new JSON Web Token JWT2 and signs this token with the registrar's private key.
5. **Registrar** calls the DIDInventory smart contract as message sender and passes the device address. This step is needed since the blockchain should not handle JWTs due to their length and the resulting gas costs. DIDInventory registers the assignment between device and registrar with a tentative state.
6. **Registrar** submits JWT2 to a **MASA-BI** node which is a server application connected to the blockchain.
7. **MASA-BI** node checks the validity of JWT2 and the registration in DIDInventory (step 5). If both are valid the MASA-BI node proofs the assignment in DIDInventory. Afterwards DIDInventory changes the state to active.
8. **MASA-BI** node generates a JSON Web Token JWT3 with a confirmation about the assignment, signs it with its private key and sends it to the **registrar**.
9. **Registrar** forwards JWT3 to the **device**.
10. **Device** verifies the signature of the **MASA-BI** node with its built-in list (in secured environment) and if ok adds the registrar to its trust list.

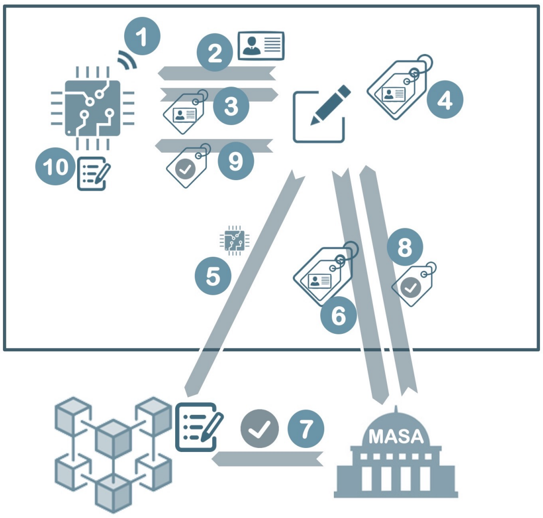


Figure 2: Bootstrapping process UC4.

As extension the use of a nonce can be applied to enhance security (see [1]). Furthermore, the JWTs could be provided with an expiration time to reduce the risk of a replay attack.

Since the DIDInventory holds the assignment the registrar can always check this using the read function. This function is restricted to the individual registrar. We are aware that at the actual prototype using the transaction history everybody can possibly read this assignment. It is our intention to improve this in a second version with the actual developments going on regarding Zero Knowledge Proof and Ethereum 2.0.

Bootstrapping variations

Further use cases are exceptional cases and are based on UC4:

UC5: Assignment of a device that is out of reach of an internet connection

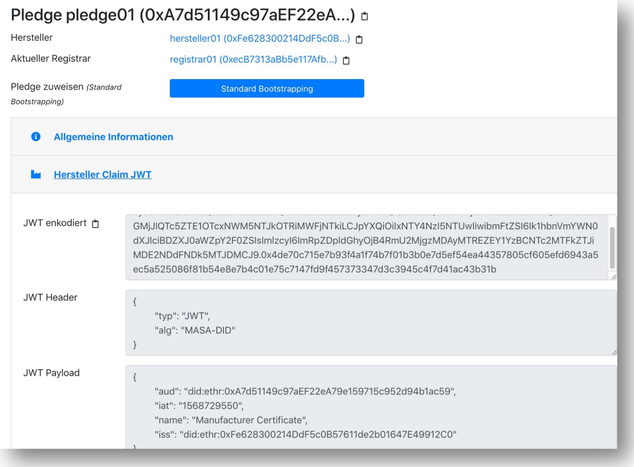
UC4 assumes that the device, the registrar and the MASA node are connected. If the device is placed in a shielded environment where no direct internet connection is possible the registrar can act as a transportation medium. In this case the registrar has to move from the shielded environment of the device to an environment where an internet connection is possible. The expiration of the JWTs has to be chosen accordingly.

UC6: Transfer of a device from registrar A to registrar B

There might be the need for a change in registrars. This can be the case due to change in ownership or responsibility like change of tenants or due to additional registrars.
In our approach this case is handled by a two-phase process. In a first phase the assigned registrar A reports a new registrar B to DIDInventory. In the second phase UC4 is applied to registrar B and DIDInventory handles the transfer. We use a special type attribute in the JWT payload to indicate the device that no reset of its settings should be performed.
The first phase of UC6 can also be used as backup of a registrar and is time-independent from the second phase.

6. Prototype implementation

We used the Ethereum blockchain for a technical implementation of the prototype with Solidity as developing language for the smart contracts. The use cases have been implemented separately so they can be easily transferred to an infrastructure with multiple devices. In a first step we realised a software prototype with all use cases as single components in a Javascript Node environment with a React frontend. This test environment allowed a step-by-step verification of the described use cases and validation of all sent and received information (see Figure 3).



Pledge pledge01 (0xA7d51149c97aEF22eA...)

Hersteller: hersteller01 (0xFe628300214DdF5c0B...)

Aktueller Registrar: registrar01 (0xecB7313aBb5e117Afb...)

Pledge zuweisen *(Standard Bootstrapping)*: Standard Bootstrapping

JWT enkodiert:

```
GMjJlQTc5ZTE1OTcxNWM5NTJkOTRiMWFjNTkiLCJpYXQiOiIxNTY4NzI5NTUwIiwibmFtZSI6Ik1hbnVmYWN0
dXJlciBDZXJ0aWZpY2F0ZSIsImlzcyI6ImRpZDpldGhyOjB4RmU2MjgzMDAyMTREZEY1YzBCNTc2MTFkZTJi
MDE2NDdFNDk5MTJDMCJ9.0x4de70c715e7b93f4a1f74b7f01b3b0e7d5ef54ea44357805cf605efd6943a5
ec5a525086f81b54e8e7b4c01e75c7147fd9f457373347d3c3945c4f7d41ac43b31b
```

JWT Header:

```
{
    "typ": "JWT",
    "alg": "MASA-DID"
}
```

JWT Payload:

```
{
    "aud": "did:ethr:0xA7d51149c97aEF22eA79e159715c952d94b1ac59",
    "iat": "1568729550",
    "name": "Manufacturer Certificate",
    "iss": "did:ethr:0xFe628300214DdF5c0B57611de2b01647E49912C0"
```

Figure 3: Frontend screenshot of the software prototype.

In a second prototype, we are building a hardware-based system with separate components for device, registrar and MASA-BI. To experiment with different hardware configurations and transmission protocols we use Arduino and Nordic NRF52840.

One of the important aspect of using Ethereum is the cost for transaction and execution of individual steps. If the mainnet of Ethereum would be chosen, the cost of about 4Mio Gwei[9] per registration would arise. This would result in costs of 2,57 CHF[10] which is too much for an industry usage. Therefore, we suggest the set-up of an own Ethereum[11] network run by different manufacturers and organised as association. This would allow the usage of a Proof of Stake consensus mechanism and the independence from highly volatile crypto prices.

7. Strengths and weaknesses of such a system

The proposed system offers a variety of benefits for both manufacturers and customers. These are not only based on the usage of blockchain technology but also on the application of the chosen identity solution using DIDs. Nevertheless, there might also be some drawbacks. We analyzed strengths and weaknesses from a stakeholder perspective. This analysis is without claim to completeness.

7.1. Benefits for the manufacturer

Device inventory
Today most manufacturers have to keep track of their produced devices by an own infrastructure. The first part of our solution (UC1 and UC2) can substitute this with an immutable and distributed ledger offering an audit trail on all devices. Since we designed the system that those use cases could also be separated in an own blockchain infrastructure, any concerns about showing numbers of devices produced can be dispelled. It has to be mentioned that if there is a separation between the identity providing and bootstrapping no further verification about the device origin is possible in DIDInventory during bootstrapping.

MASA-BI ecosystem
Our vision is an open, community-oriented ecosystem for the MASA-BI infrastructure. This community-supported MASA-BI would facilitate an open and transparent market. For start-ups this would also make it easy to participate in a secured device distribution. From this open ecosystem all market participants could benefit. To ensure the open character and to prevent a takeover by one market player, an association or foundation as legal form is suggested.

Security
Device security today is mainly based on certificates from CAs (Certified Authorities). Assaults on those CAs and disclosure of root certificates result in a massive security issue for all devices trusting in those certificates. Self-sovereign identity of devices and the proposed blockchain-based approach reduce this risk significantly.

Proof of origin
Since the devices are registered during manufacturing, a proof of origin and trusted supply chain can be guaranteed. If devices are traded on a secondary market the history of those devices can be retraced. For some environments like critical infrastructures second-hand devices are not allowed.

[9] Using an average Gas price of 55Gwei (23.12.2020)

[10] Ether price of 648 CHF (30.12.2020)

[11] E.g Hyperledger Besu or Quorum

Our approach is a way to detect such misuse. Even if a device is used and not assigned to a registrar, a factory reset can be enforced. In addition, the exact manufacturing date can be reconstructed from the registration time on the MASA-BI.

Firmware update
Finally, the system could be extended to a registration of the registrar at the manufacturer. This identification should be separated from the MASA-BI due to GDPR reasons. A direct link between the manufacturer and the registrar could simplify sending firmware or factory updates regarding the specific device versions. Linking registration to MASA-BI and registration with the manufacturer is one way to increase the registration rate of devices.

7.2. Possible drawbacks for the manufacturer

Transparency about production
In a full extension where device registration at manufacturer site (UC2) and bootstrapping (UC4) are handled by the same permissionless blockchain, it will be possible to draw a conclusion about the number of produced devices. For some manufacturers this might be a problem. The further development using zero-knowledge proofs or permissioned blockchains can eliminate this obstacle. Nevertheless, some manufacturers could be restrained.

Costs
Registration of devices at the MASA-BI is associated with costs. The prototype is using the Ethereum blockchain where Gas has to be paid for writing transactions. On a large scale these costs can sum up to a significant amount, especially with recently rising Gas costs. There are several possibilities to solve this drawback. The infrastructure could be provided by an independent association or the nodes of the blockchain used can be financed by different manufacturers. With this approach of a permissioned blockchain infrastructure a new way of pricing can be implemented.

7.3. Benefits for the customer

Easy registration
The registration process for new devices should be as convenient as possible. The proposed system facilitates this reduction in complexity. Since there is no constraint to follow the registration, the benefits for the customer should nudge him to use the system. This feature is very much dependent on the usability of the registrar software. Therefore, special attention must be paid to this.

Security of device origin and counterfeit discovery
Especially for commercial usage the origin of a device is decisive (see 7.1 – Proof of origin). Due to a transparent tracking, the proposed system allows to detect irregularities in the supply chain. Not only do manufacturers benefit from this, customers benefit as well, as the tracking of devices is possible without involving manufacturers.

Fallback scenario if registrar is changed
To enhance convenience, all situations where a registrar is involved have to be considered. UC6 already addresses these aspects. We are working on further processes to cope with this scenario. Again, usability and security are the main focus.

Keep configuration even if complete system is handed over to another provider.
In an environment where a service provider is responsible for the setup and configuration of a system, a handover to the operator is required. UC6 addresses this handover and raises the convenience level. This is a great opportunity since today installations have to be set up in a new way if a handover happens.

7.4. Possible drawbacks for the customer

Transparency about device ownership
Our proof of concept uses Ethereum as blockchain technology. The open character of this blockchain allows conclusions about the ownership of devices registered. This might be a similar drawback for the manufacturer (see 7.2). Further development in blockchain technology as well as access restriction to data can cope with this drawback.

Need for having a registrar
The implementation of the proposed system requires the usage of a registrar for each installation site. For smaller sites this might be a dissuasive effort. Therefore, it is required that the effort for setting up and operating a registrar is reduced to the minimum. Nevertheless, a customer will only use this kind of system if the benefits mainly and the convenience level are high.

7.5. Summary as SWOT

Strength	**Weaknesses**
- open ecosystem - security - proof of origin - easy registration	- costs, if public blockchain - registration process needed - registrar needed
Opportunities	**Risks**
- use as inventory - manage firmware upgrades - easy handover of installation	- transparency over devices

8. Conclusion

We presented a concept for a device registration system based on blockchain technology. This system allows the allocation and management of device identities which are independent of manufacturer-provided systems. Therefore, our proposal of a self-sovereign identity management is immutable and independent of the failure of any single player.

The usage of already existing signing technologies in combination with JSON Web Tokens and the concept of DIDs allows a fast and lean implementation. Due to the application of secured hardware the access to the identity can be kept on the device. Just like identity for humans, the identity of things will be an essential feature for future applications.

Competing Interests:
None declared.

Ethical approval:
Not applicable.

Author's contribution:
TW took the lead in conception and writing of this article. OC provided industry inputs, revising and approval for the publication of the content.

Funding:
The research was funded by Siemens Schweiz AG.The funding was in no respect linked to the use of any kind of technology or methodology. Therefore, the authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgements:
The authors are grateful to Siemens Schweiz AG for supporting this research. Additionally, we would like to thank Christian Renold for the implementation of the prototype and support during the verification of the use cases.

References:

[1] M. Pritikin, M. Richardson, T. Eckert, M. Behringer and K. Watsen "Bootstrapping Remote Secure Key Infrastructures (BRSKI)" https://tools.ietf.org/html/draft-ietf-anima-bootstrapping-keyinfra-30 [Accessed March 25, 2020]

[2] T. Weingärtner "Tokenization of physical assets and the impact of IoT and AI." https://www.eublockchain forum.eu/sites/default/files/research-paper/convergence_of_blockchain_ai_and_iot_academic_2.pdf, 2019, [Accessed March 25, 2020]

[3] X. Zhu and B. Youakim "A Survey on Blockchain-based Identity Management Systems for the Internet of Things." 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), 2018.

[4] P. Kohlhaas "Zug ID: Exploring the First Publicly Verified Blockchain Identity" https://medium.com/uport/zug-id-exploring-the-first-publicly-verified-blockchain-identity-38bd0ee3702, 2017 [Accessed March 25, 2020]

[5] A. Third, K. Quick, M. Bachler and J. Domingue "Government services and digital identity" https://www.eublockchainforum.eu/sites/default/files/research-paper/20180801_government_services_and_digital_identity.pdf [Accessed December 30, 2020]

[6] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grand and M. Sabadello "Decentralized Identifiers DID (v1.0)" https://w3c.github.io/did-core/ [Accessed December 30, 2020]

[7] N. Otto, S. Lee, B. Sletten, D. Burnett, M. Sporny and K. Ebert "Verifiable Credentials Use Cases; W3C Working Group Note 24 September 2019" https://www.w3.org/TR/vc-use-cases/ [Accessed December 30, 2020]

[8] J. Shane "Welcome to uPortlandia!" https://medium.com/uport/welcome-to-uportlandia-2302e0d2ceb1 [Accessed December 30, 2020]

[9] L. Lesavre, P. Varin, P. Mell, M. Davidson and J. Shook "A taxonomic approach to understanding emerging blockchain identity management systems". arXiv preprint arXiv: 1908.00929 DOI 10.6028/NIST.CSWP.01142020, 2019.

[10] J. G. Faísca and J. Q. Rogado "Decentralized semantic identity" in Proceedings of the 12th International Conference on Semantic Systems, 2016, pp. 177-180.

[11] "Introduction to JSON Web Tokens" https://jwt.io/introduction/ [Accessed March 25, 2020]

[12] A. Mühle, A. Grüner, T. Gayvoronskaya and C. Meinel "A survey on essential components of a self-sovereign identity". Computer Science Review, 30, 2018, pp. 80-86.

[13] D. van Bokkem, R. Hageman, G. Koning, L. Nguyen and N. Zarin "Self-sovereign identity solutions: The necessity of blockchain technology". arXiv preprint arXiv:1904.12816, 2019.

[14] Q. Stokkink and J. Pouwelse "Deployment of a blockchain-based self-sovereign identity" 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), 2018, pp. 1336-1342.

[15] Y. Liu, Z. Zhao, G. Guo, X. Wang, Z. Tan and S. Wang "An identity management system based on blockchain" in 2017 15th Annual Conference on Privacy, Security and Trust (PST), 2017, pp. 44-4409

[16] C. Allen "The Path to Self-Sovereign Identity" http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html, 2016 [Accessed March 25, 2020]

[17] M. Al-Bassam "SCPKI: A smart contract-based PKI and identity system" in Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, 2017, pp. 35-40.

[18] U. Der, S. Jähnichen and J. Sürmeli "Self-sovereign Identity - Opportunities and Challenges for the Digital Revolution". arXiv preprint arXiv:1712.01767, 2017.

[19] P. C. Bartolomeu, E. Vieira, S. M. Hosseini and J. Ferreira "Self-Sovereign Identity: Use-cases, Technologies, and Challenges for Industrial IoT" in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2019, pp. 1173-1180

[20] C. Shepherd, G. Arfaoui, I. Gurulian, R. P. Lee, K. Markantonakis, R. N. Akram and E. Conchon "Secure and trusted execution: Past, present, and future-a critical review in the context of the internet of things and cyber-physical systems" in 2016 IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 168-177

[21] G. Wood "Ethereum: A secure decentralised generalised transaction ledger". Ethereum project yellow paper, 151, 2014, 1-32.