Identity management in IoT networks using Blockchain and Smart Contracts

ABSTRACT

The penetration of usage of blockchain in Internet of Things is happening in managing smart lockers in shared economy sector, in decentralized smart solar energy meters and in devices peer-to-peer communication. The IoT devices identity management also is a domain that we here introduce how it can be implemented using blockchain technology. Blockchain features of immutability, provenance and cryptographic infrastructure provides a platform to implement functions in IoT devices identity management to ensure that devices can hold global and unique identities that can be maintained throughout the device lifecycle and can be traced and audited to the birth certificate of each device. We here present a semi-decentralized blockchain-based identity management framework that provides features of identity creation and transfer of ownership. We demonstrate the work by a use case that provides a tool for mobile phone counterfeiting where we deploy that on the Ethereum blockchain testnet and implemented using smart contracts.

Index Terms: **Identity Management, Internet of Things, Blockchain, Smart Contracts**

I. Introduction

In the Internet of things IoT, the role of identity management is expanding not only to include users' identifications and managing their access to different types of devices and data, but it is addressing how to identify the devices and manage the interaction between devices, their offered services and the access to sensitive data. Such a change is requiring a functionality of the Identity and Access Management IAM to be capable of managing human-to-device, device-to-device, and/or device-to-service/system.

Furthermore, in many cases IoT devices are connected intermittently and/or temporarily and are required to communicate with other devices, services and the infrastructure [1]. During this, the authentication of those devices and authorization given should be performed according to a well governed identity management lifecycle that can be applied to the devices and ensure unique and global identification mechanism. This yet will be constrained by ensuring a considerable security levels in terms of authenticity, auditability and access rights through the identity management lifecycle. This includes, establish clear registration processes for IoT devices, maintaining global and unique identity and providing the frameworks for access control and authorization.

The other factor is the increasing number of devices and heterogeneous relationships among them, which is becoming more popular in both the consumer and the enterprise domains. Though, still the vast majority of IoT platforms are based on a centralized model in which an organization authority handles all the interaction between devices and with other devices belonging to other authorities [2]. However, many impractical issues in this approach are mounting; especially in the scenarios in which devices need to exchange data between themselves independently, in a peer-to-peer fashion. This specific requirement has led to ontroduction of the decentralized approach. The blockchain technology as a shared and distributed ledger can facilitate the implementation of decentralized IoT platforms that can maintain the secured and trusted data exchange as well as history and records keeping based on its immutability feature and anonymity of its identification scheme. In such an architecture, the blockchain serves as the general ledger, keeping a trusted record of all the transactions and events exchanged between smart devices in a decentralized topology.

This work proposes utilizing the blockchain technology in addressing identity management in IoT, to bring a solution for establishing a unique and global digital identity that can be maintained throughout the device lifecycle, in addition to supporting the device ownership management and identity update. The ultimate objective of using blockchain is to transition identity management to a decentralized platform in IoT. Though, we focus in this paper on employing the main concepts of bloackchain that are based on public and private keys for identification, the distributed ledger and its immutability and provenance for identity management through devices' lifecycle and smart contracts for process and rules execution.

The paper is structured as follows. In Section II the concept of identity management is described with mention of suggested and used approaches, then we introduce what a blockchain is, how it operates, its main features, and how smart contracts can be used to implement the processes and rules governing the interactions between the devices. Section III provides a look into the suggested framework for adoption of blockchain, and highlights the suggested applications. We describe the details of implementation and use cases as well as the main issues and points to consider when deploying a blockchain-based solution in IoT in Section IV. Finally, Section V presents the conclusion and summary on the implementation results.

II. Identity Management in IoT

Identity management is an administrative domain that deals with identifying individual entities in a system (such as a user or a device) and controlling their access to resources within that system by associating access rights and restrictions with the established identity [3]. This is usually described as authentication and authorization and a broader scope of the definition also includes the identity creation and maintaining it through lifecycle.

One of the major IAM approach is Identity Relationship Management IRM backed by the Kanatra Initiative [4], which is considered as a main stream, yet many challenges exist in this stream as it needs to handle the many relationship models that need to be built and maintained. For example, the issues of providing unique and long-term identity and addresses in a global sense requires an infrastructure in place that supports highly dynamic devices that join and disjoin the network at any time and move between different owners and networks.

Multi-Factor Authentication MFA and Authentication and Authorization for Constrained Environments ACE [5] are also standardized approaches that focuses on authorization and authentication for constrained devices to ensure that miniature IoT devices can still maintain the required level of security.

In [6], Chen et al. suggest an IoT user-centric Identity Management framework, they introduce the difficulty of a centralized approach and show the information model and relationship between entities, identities and attributes. Their user-centric IdM architecture for IoT is built on a global identity provider that is responsible for maintaining global identity while different service providers generate local identity according to global identity, and keep consistency of information with global identity. Thus, each local identity may has its own credential and authentication method. The proposed framework does not differ from a federated identity management framework, and does not show how the global identity and local identities are maintained, generated and orchestrated.

Trnka and Cerny [2], propose a centralized IoT identity management framework, the work is based on generating a unique global identity maintained by a central store. The whole identity management and authorization occurs in the central store, and all communication initiation and service access is authorized based on OAuth 2.0 token generation governed by the central store. The work does not address the scalability issue or the intermittent access of IoT devices. Ther is also no mechanism on device identity management through the lifecycle of the device, and no mention on how transfer of ownership can be done in silo isolated identity stores.

Blockchain technology has gained interest lately in defining a decentralized Identity management frameworks for IoT. IBM in partnership with Samsung has developed a proof of concept platform ADEPT (Autonomous Decentralized Peer-to-Peer Telemetry) that uses elements of the bitcoin's underlying design to build a distributed network of devices as a decentralized Internet of Things (IoT). ADEPT uses three protocols; BitTorrent (file sharing), Ethereum (Smart Contracts) and TeleHash (Peer-To-Peer Messaging) in the platform. The PoC shows the capability through smart contracts on Blockchain to automate simple functions such as registration and authentication [7].

Chronicled Inc. also is providing an Open Registery for IoT where encrypted microchips (currently only Bluetooth low Energy and NFC) are used to assign a secure digital identity to a physical object and linking it to a blockchain record. The solution also provides a verification application on a mobile phone to scan a product and verify it identity. Chronicled showed its technology through a sneakers and apparel use case [8].

Based on this, the main points of interest that we address in relation to the IAM module of IoT management framework include:

- Define a global and common registry for IoT devices.
- Build the necessary identity lifecycle that can be applied to things in an organization and can be tailored based on the lifetime of the device.
- Define clear registration processes as the start of the IoT devices lifecycle.
- Ensure applying security measures during the registration process due to the fact that compromising identity of the IoT device compromises the sensitive the data handled by the IoT device.
- Provide the means to handle devices ownership management and transfer of ownership.
- Provide the mechanism to trace the lifecycle of the device back to the origin.
- The creation of device digital identity and the need to reflect the device lifecycle changes in the identity formation.

III. Blockchain and Smart Contracts

Blockchain technology has lately received a lot of interest in different industries, especially in financial services. While the first application of blockchain technology are crypto-currencies namely Bitcoin, that have become an alternative to commodity money; however, over the past few years an entire ecosystem of new companies has developed different blockchain applications. These applications brought new kind of decentralized, secure and fast means of handling domain-specific transactions; in healthcare, postal service, IoT, supply chain and others.

As a technology, Blockchain is defined as a distributed shared digital ledger for transactions, which employs public key cryptography in establishing the identity and pseudo-anonymity of all participants and decentralized consensus algorithms to maintain the intactness of the ledger and verify any transaction. The distributed ledger is made through combing a number of transactions in a block, hash their content with the hash value of the previous block to construct the hash value of the new block. This happens for each newly generated block except for the genesis block.

1. The chain structure

One of the main important features of blockchain is immutability where transactions that are recorded such that they are irreversible and permanent. The change, for example, in the balance of an account is not done through updating a balance field, rather each new transaction is combined with other transactions are used to form a block that will be validated and distributed as well as linked to the previous block. Blockchain name can be literally described by its two key words; firstly that each (n) group of transactions form a single block, the second is to link blocks via cryptographic hashes into an immutable chain, such that it is easy to verify the historical record down to the first block (the genesis block). This in addition of ensuring immutability it offers the feature of provenance [9].

The blockchain secure workflow can be described in the following steps:

1. An external entity that owns a pair of private/public keys and a blockchain address, which is a short format of the public key, sends transaction through a blockchain node.
2. The transaction is cryptographically signed by the entity private key, while the blockchain will identify the entity through its address (public key) in a pesduo-anonymous way.
3. The transaction is validated and with other transactions occurring in a period of time are ordered and grouped to form a block.
4. The generated block is used to challenge the mining nodes, the first to solve the challenge generates the so-called proof-of-work (PoW) and broadcast the block to the rest of the network.
5. The rest of the networks confirm the validity of the transactions and if confirmed it inserts it into the chain by linking it to the previous block, and all of this occurs according to the consensus algorithm.
6. Global image of the shared distributed ledger now is updated and in case of financial transaction the recipient receives the transaction.

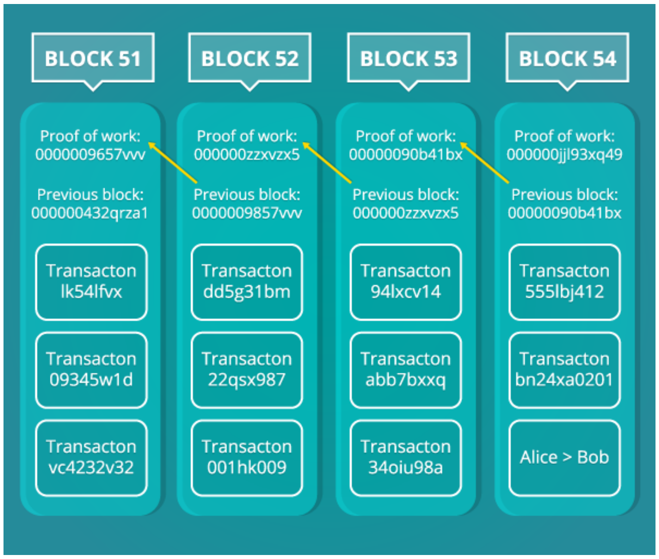


Figure 1 - The chain of blocks

2. Ethereum and Smart Contracts

Ethereum is a blockchain architecture with an associated state database, capable of storing programs and their state. These programs are commonly referred to as Smart Contracts. A smart contract can be deployed by any Ethereum user and it has a function-based interface or Application Binary Interface ABI. Once deployed the smart contract can be referenced by its address, which is a cryptographic identifier. A user can call a smart contract function by sending a transaction with this address as the destination, and with the data payload of the transaction containing the function signature and input parameters. Calling a function causes the miners of the network to execute the program and update its state. A smart contract can hold and send the native value token Ether, and can furthermore call functions of other smart contracts [10].

Ethereum blockchain provides a “Turing complete” coding system, where theoretically code can put with any logic into an Ethereum smart contract, and it will be run by the whole network Ethereum Virtual Machine EVM. The EVM computation capabilities is prevented from abuse, by allocating a token (Gas) against each computation so computation power is paid for. The ecosystem of Ethereum blockchain-based solutions is made of the Ethereum network composed of number nodes (can be either public or private), smart contracts that are implemented in one of the supported languages (Serpent and Solidity) and user interface through other libraries [10].

For the purpose of clarity in the next sections, we introduce in the table below few terminologies that are used in the smart contract programming, which will be used in the text for the details of the implementation.

IV. Semi-Decentralized Identity management in IoT

As mentioned in the introduction, industry trends aim for a decentralized architecture in IoT, which is seen essential in the service layer. However, for the underlying management functionalities, we suggest a semi-decentralized architecture that can be realized while it can enable decentralized architecture for IoT devices communications and services. In this paper we introduce a semi-decentralized identity management framework for IoT, this is based on the blockchain and smart contracts technology. The framework highlights three main principles: identity as a living asset, identity being unique and global and minimal third-party authority.

1. The concept of Identity for IoT devices

The need for identity management is essential in all IT enterprise solutions to handle all the authentication and authorization flows; yet in IoT this need is driven further by factors related to the high number of devices, complex architecture, change of ownership and intermittent network association. This requires building a global and unique identity of devices that can be transferred between networks and owners with less impact on its identification and optimally with no loss of track of the historical records.

In this work, we define identity as a living asset, such that a device is given its identity at birth (e.g. manufacturing). The device will maintain its identity with updates and versions as pre-defined changes happen, which will be maintained in a global registry to enable tracking and auditing. According to the ISO/IEC 24760-1 [11], an entity is an item inside or outside an information and communication technology system, such as a person, an organization, a device, a subsystem, or a group of such items that has recognizably distinct existence, while an identity is the "set of attributes related to an entity".

In this work we adopt those definitions in a relationship model between entities (devices), identities and attributes that is built on one-to-one relationship between the device and its identity and one-to-many relationship between an identity and the composing attributes. In addition, we require that an identifier is essential that should be unique, global and optimally anonymous.

Thus, for a system with (n) entities, there exists (n) identities each with a unique identifier ID, the identity is defined over a set of attributes **A**

$$E_i <= ID_i <= <A_{i1}, A_{i2}, \ldots, A_{im}>$$

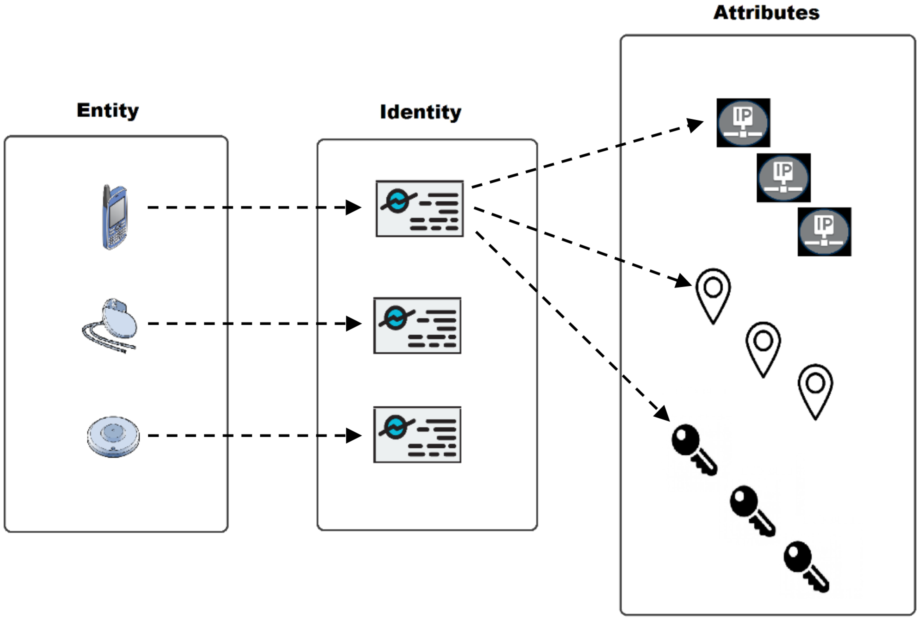


Figure 2 - Entity-Identity-Attributes Relationship Model

2. Device Identity lifecycle and Ownership

In the IoT networks the device will go through a lifecycle starting from manufacturing till end of life. The suggested IDM framework depicts a flow that defines the device status and its ownership. In this lifecycle, the device identity is maintained through a global registry that permits authorized registrants (e.g. manufacturers) of creating new identities for new devices, and either perform identity management functions or transfer ownership of the device(s) to operators/users. The identity as a living asset is maintained and updated through the different phases, where a new version if required is issued as the device's owner changes or the device status changes from provisioning to in-service to retire.

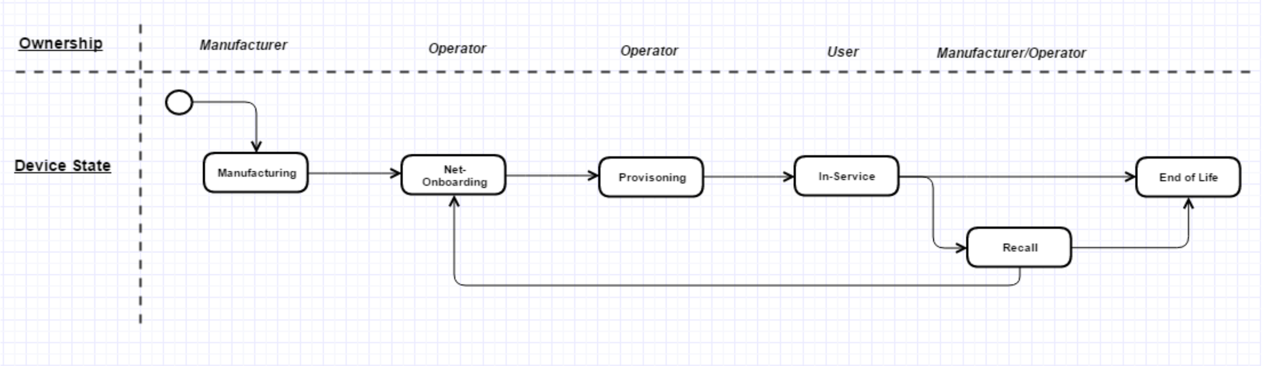


Figure 3 - IoT Device Lifecycle and Ownership

V. Blockchain-based system architecture

The essence of architecture is based on the main concepts of blockchain that we use to match the objectives of establishing a global registry with unique and global identifier for each device, with higher degree of anonymity, with immutable records, traceable to the genesis of the identity creation and with minimal third party authority.

The distributed ledger of blockchain is used with smart contracts on top to build the global registry with specific rules to enable authorized and identifiable entities of the creation of the device identity. The public key cryptography is used as the mean to ensure establishing a global and unique identifier for each device. In Blockchain and specifically in Ethereum, each account (e.g. device) will first have a private key and public key generated, then produce a blockchain address based on result of the hashing function of the public key, those details are described in the next subsections.

The high level system architecture defines the main blockchain components and the different stakeholders and how they access and utilize the system.

1. Identity creation

The identity creation is based totally on the public key cryptography and hashing functions. The identity is composed of the minimal set of attributes necessary to generate a unique and global identity for the device in the global registry. That is performed in two steps: generation of cryptographic keys and blockchain address and then blockchain-based digital identity formation.

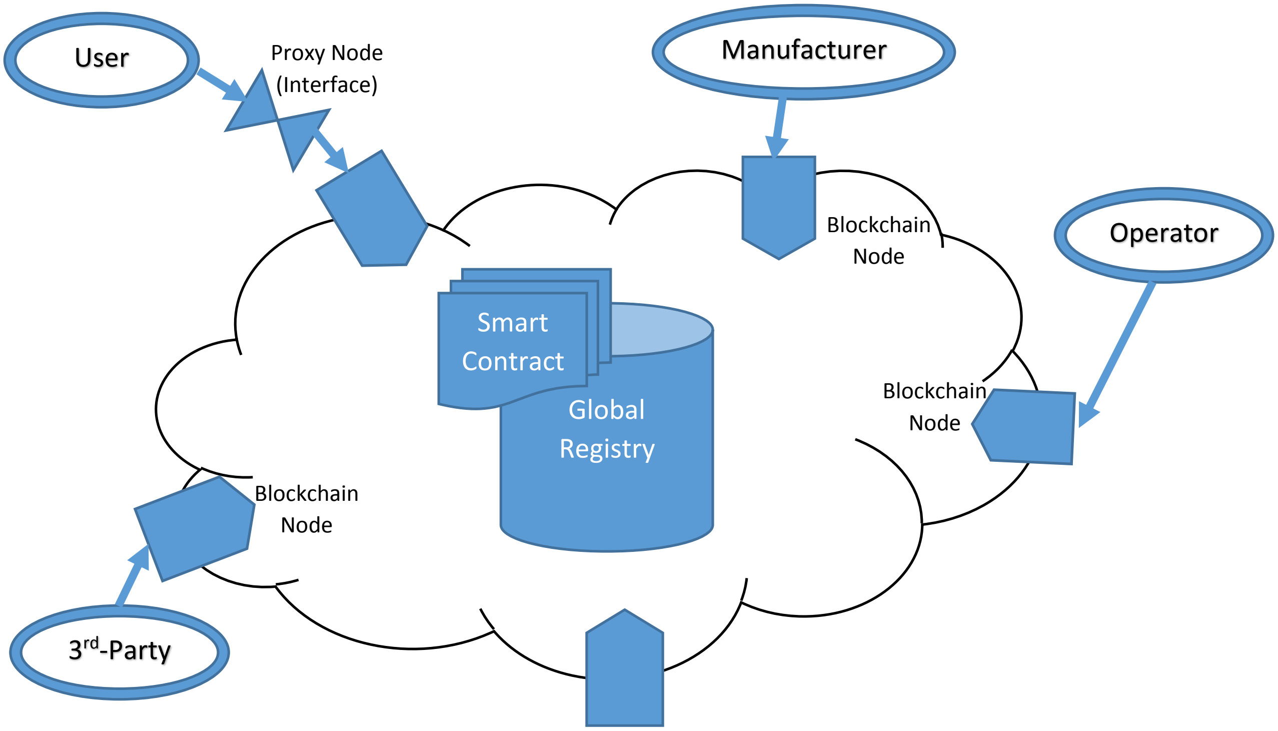


Figure 4 - High Level Overview of System Components and Stakeholders

i. Private and public key generation

Devices start their lifecycle at the manufacturer's facility, the suggested approach will mandate the manufacturer to create the private and public cryptographic keys for the device, and extract the (Ethereum) blockchain address based on the public key. We initially suggest that those keys pairs are then certified and issued a device certificate that should be loaded into the device with the signer (manufacturer) certificate. The details of the key pairs generation and Ethereum address is given below [10]:

1. An initial seed vector (IV) is used with a key derivation function such as PBKDF2-SHA256 to generate a 256-bit (32 bytes) private key (SK).
2. The 512-bit (64 bytes) public key (PK) is generated from that secret key (SK) using the elliptic curve digital signature algorithm (ECDSA).
3. The Public Key is hashed with SHA-3 (Keccak) to produce a 256-bit output, where the upper 96 bits are discarded, and the lower 160 bits become the Address.

$$A(SK) = \beta_{96..255}(KEC(ECDSAPUBKEY(SK)))$$

ii. Device Digital Identity on Blockchain

The unique and global identifier represented by the blockchain address that is extracted from the device created private-public key pair is used as to issue the birth certificate of the digital identity of the device through the deployed smart contract for identity creation and ownership management.

The smart contract for identity creation requires each device to have its minimal set of attributes hashed using a Keccak hashing function [12] to produce a 20 bytes digital profile. This hashed value will then be signed using the device private key to ensure its validity, and submitted to the smart contract interface. The minimal set of attributes can include device serial number, device manufacturer name or product name or MAC/IP address.

Using a two-factor identification, both the Ethereum address and the attributes hash are hashed with a timestamp based on the current block to generate an internal index in the smart contract. This will be used as the main index to store the device digital identity in the blockchain global registry. The digital identity will contain the digital profile of the device based on its set of attributes, the unique global identifier representing its address, the owner global unique address, the identity version to reflect any updates, the block number and a flag to indicate certain permissions on the device identity such as being transferable, assignable, updatable and retractable.

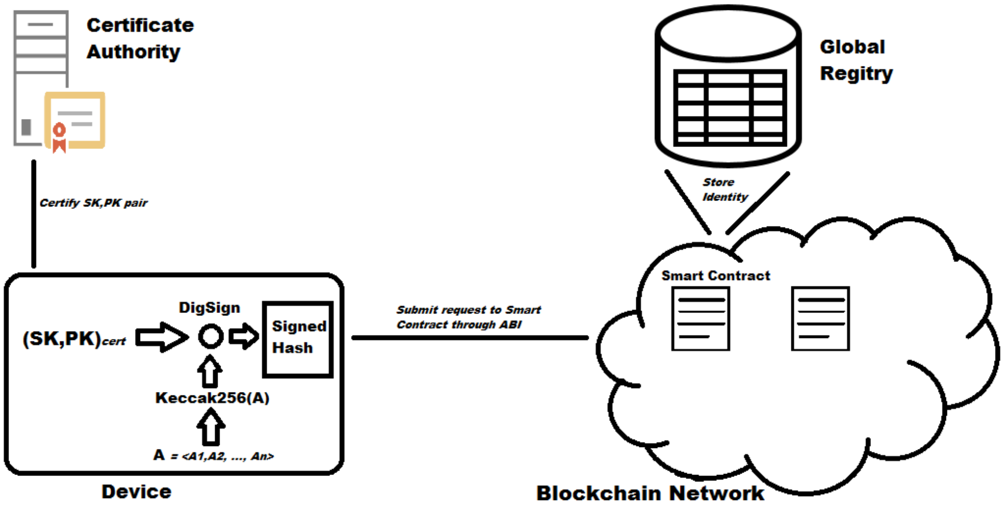


Figure 5 - Identity Creation and Registration System Overview

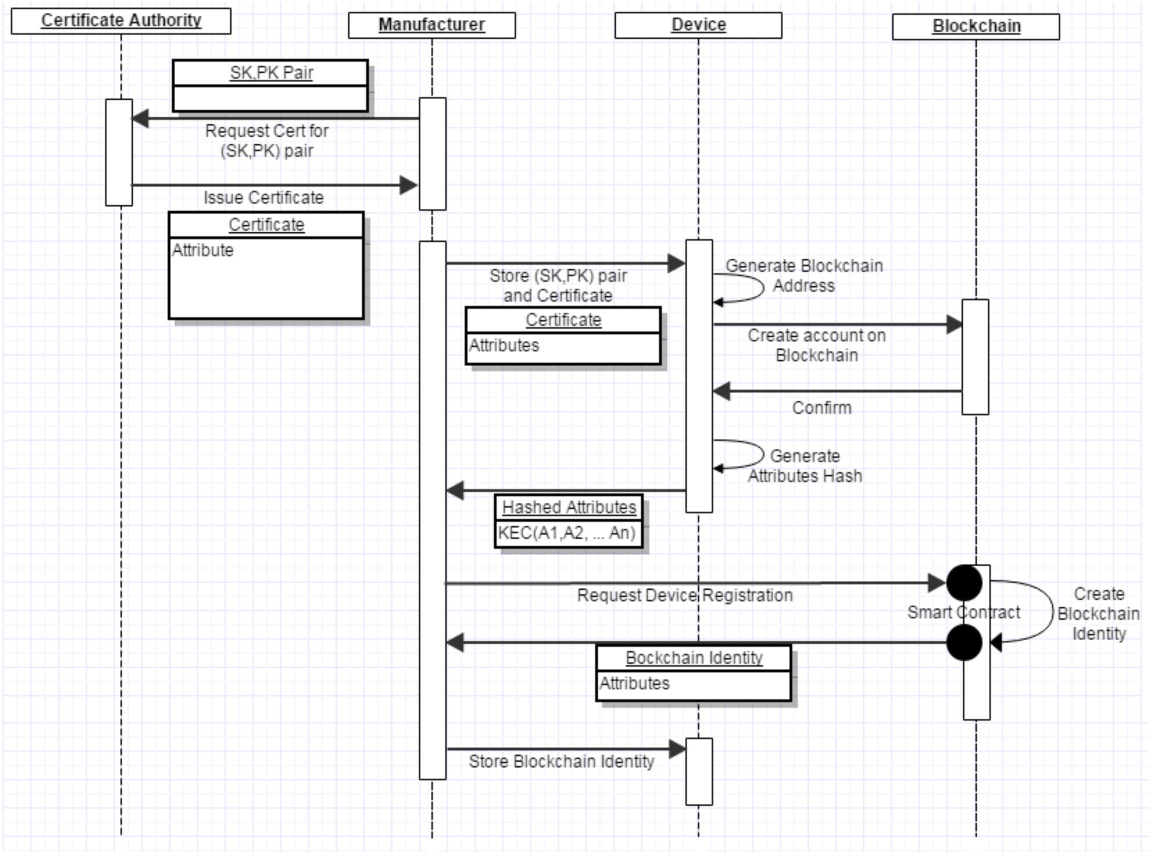


Figure 6 – Detailed Process Flow for Identity Creation

2. Identity Ownership

The deployment of IoT devices goes through the aforementioned steps of the lifecycle. Those steps define the owner of the device and dictate the transfer of ownership. Handling identity management in IoT brings a great attention to the ownership management, not only in regard of reflecting the actual asset status and the related supply chain management, but even more is the security and trust concern. According to the IoT security foundation [13], it is inevitable to limit IoT devices from changing the ownership through their lifetime, and the essential task in ownership management is to preserve the security of the device and data throughout its lifecycle. Thus, as devices are transferred from one owner to another, the identity management framework should sustain:

- *Secure mean of transfer of device identity ownership from one user to another.*

  Transferring the device ownership will require performing the transfer process in a secure manner that ensure protection of device sensitive data.

- *Cross-ownership permissibility*

  In certain applications, the device ownership can be split between different owners, e.g. an IoT device in a smart home scenario can be owned by the tenant as for its data, but the hardware is still owned by the lender or the operator.

- *Change of ownership should not affect the device security updates*

  The ownership of the device and any transfer should be agnostic to the security operations and processes.

In addition, in this work we add two other requirements on IoT device identity ownership and transfer of ownership, and these are;

- *Ownership tracing*

  As devices' ownership changes through the lifecycle and to ensure the capability of auditing and tracing any device data compromise and also to enable construct the device reputation, devices ownership through the lifecycle and all transfers should be traceable to the origin of identity creation.

- *Identity Ubiquity and Uniqueness*

  While devices change of ownership of their identity, the devices identity shall still maintain its presence in the system with uniqueness and ubiquity.

i. Blockchain approach toward transfer of identity ownership

The abovementioned requirements on device identity ownership management are approached in this work through blockchain and smart contracts. The identity ownership is ensured to accommodate the requirements by adopting the immutability and provenance features of blockchain and also using the digital identity introduced in the previous section.

The blockchain identity created on the time of device birth is built with a set of attributes that includes the blockchain addresses of the device itself and the original owner (manufacturer) in addition to the other attributes. The generated identity stored in the distributed ledger contains number of features, and the most important in this aspect of the work is the owner address. The ownership is identified by the owner blockchain address, and through the smart contract constructs this feature is limited in access only to the owner. Therefore, a transfer of ownership can only be performed by the device identity owner, which ensures secure transfer of ownership from one entity to another. Moreover, the change of ownership is based on replacing the previous owner blockchain address with the new owner address, and that can ensure agnostic behavior of device to security updates.

In regard of cross-ownership, in this work we suggest the replacement of the concept of cross-ownership as per the IoT Security Forum to device assignment, since the objectives of cross-ownership can be still served through the concept of device assignment and addition to access rights and policy. The reason for that is that cross-ownership can be applied for long term device transfers and would require clear boundaries between the owners of what they own and what they can perform. Defining what each user or owner of the device can perform is exactly served by the access policy definition, which we don't discuss in this work. Furthermore, device assignment will permit dealing with situations of intermittent connections without a change in the device ownership feature. This in case of cross-ownership will require two steps of device identity ownership transfer.

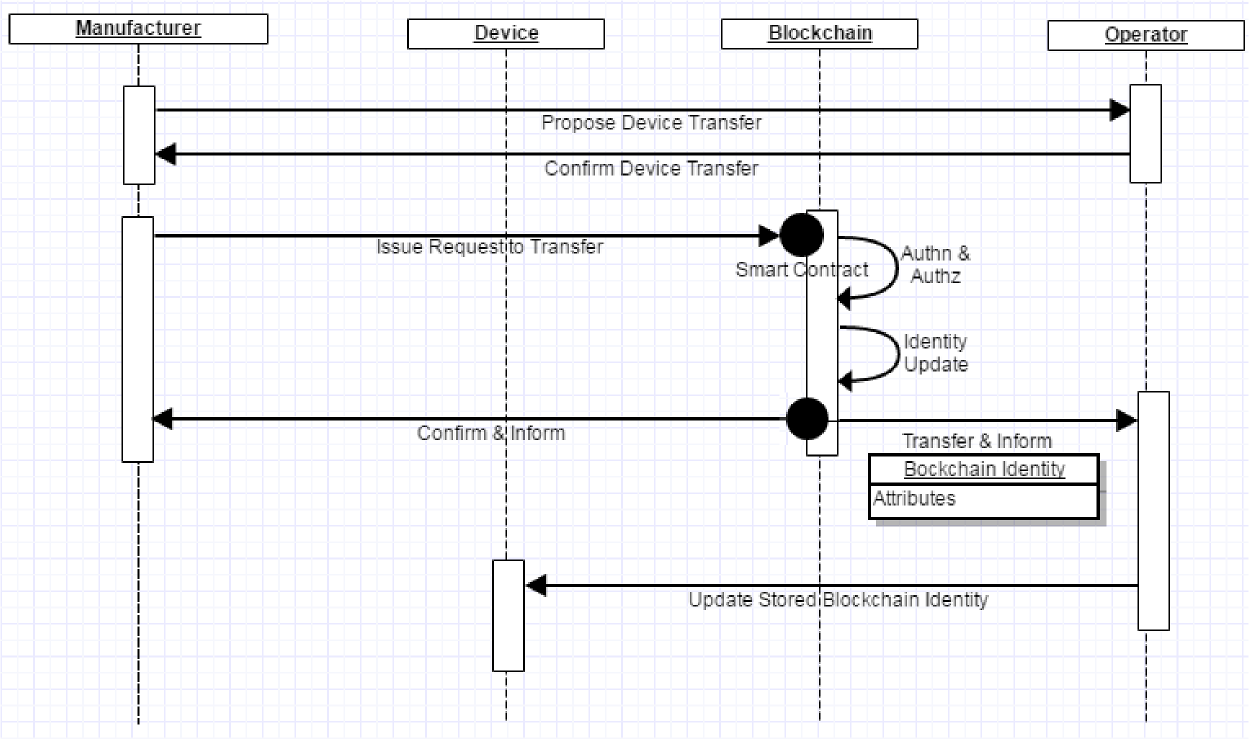


Figure 8 - Detailed Process Flow for Transfer of Identity Ownership

ii. Traceability and auditing

Based on the blockchain immutability and provenance features, all transactions are stored in the distributed ledger and those are immutable and can be tracked to the genesis block. Moreover, in this work, the blockchain identity created is formed of a data structure that contain the following features described in the figure below.

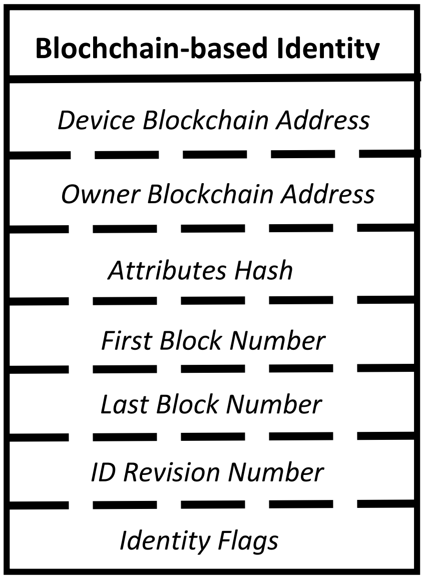


Figure 9 – Digital Identity Data Structure

Through the features of first block number and last block number in addition to the underlying blockchain functions, the identity of a device and its ownership transfers can be traced to verify its originality and intactness. The list of all identity updates and transfers are recorded in data structure on the distributed ledger that will use each ID revision number to store the block number that the transaction was stored in, which adds a layer of a virtual chain to enhance access to the ID trace.

3. Portability through A global and Unique Identity

One of the main aspects that this architecture focused on is the global and unique identity of the device in addition to considering it as a living asset. Balancing these two possibly contradicting aspects is proposed in this work through the blockchain underlying technology and the suggested blockchain-based identity. This proposal offers a global and unique identity that is ensured based on the cryptographic assumptions that is it is unique, while treating this identity as a living asset is implemented through revision control and block numbers preservation.

i. The BOYDID Concept

The proposed digital identity as a global and unique yet maintainable (through updates and revisions) is a candidate for identity portability. The portability of identity justification is rooted to how devices (entities in general) can maintain its constructed identity as they migrate from one system to another. This is a very crucial case in IT enterprise networks in general and more critical in IoT. For example, the BOYD (Bring-Your-Own-Device) model is one of the main models that would rely much more on portable identity.

In this work, based on our suggested global and unique device digital identity, we propose the concept of BOYDID (Bring-Your-Own-Device-IDentity). This concept is suggested by many in the industry for IT networks for users (humans) to enable porting a single identity of users as they transfer from one enterprise to another or switch from one service to another. While those approaches utilize open and standard authentication protocols such as (OAuth2 and OpenID)[14],[15], yet those approaches rely mostly on social logins to simplify the user authentication process. In other cases, the approach is mainly to allow a single identity sign-on scenarios, but without identity portability.

In the suggested approach, the underlying blockchain infrastructure and the uniqueness of the digital identity empowers the identity management framework to offer identity portability. The scenario can assume both a single blockchain network and federated blockchains. The BOYDID concept is based on the global and unique identity and the main identifier represented by the cryptographic keys and blockchain address. These identifier(s) will permit that devices identities can be transferred between entities and can enable carrying the device reputation as it gets transferred.

In a scenario of a wearable device brought by an employee at work, the typical approach will be mainly through the reliance on the enterprise system to detect the device and authenticate and authorize it. The global identity and BOYDID can enable IT enterprise system to pull (based on the global identifier) the security profile of the device and check it against policy rules. In other cases, such as rented home thermostat or rented connected vehicle, the manufacturer will create the original identity of the device, upon transferring the device to the operator and then to an agency or a user on frequent basis the BOYDID and the global identity will allow the system to maintain a coherent view on the device identity, its security profile and its reputation.

VI. IoT Identity Management Use Case

In this section, we present a use case of blochchain-based device identity and we demonstrate a simple application that provides the mean to users, distributors, network operators, IP protection agencies and device manufacturers a tool to combat mobile phone counterfeiting.

In Feb 2017, a report released by the European Union Intellectual Property Office (EUIPO) in conjunction with the International Telecommunications Union (ITU) reports that the effect of counterfeiting on smartphone sales worldwide is estimated at 184 million units, valued at 45.3 billion EUR or 12.9% of total sales[16]. The mobile phone counterfeiting in addition to its economic impact it has serious security, privacy and even safety concerns.

The unique identifier for any mobile phone is the IMEI number, fake phones either do not possess that ID or use a fake one. The device registration process in mobile networks for many reasons do not perform any further verification on the IMEI number, and a single centralized database is maintained by the GSM Association is used to query on counterfeited devices.

The use case implements a distributed blockchain-based registration, query and verification of counterfeited phones. The open global registry is used to store the unique cryptographically generated digital identity, this identity will be generated by the manufacturer for each device using IMEI as the main attribute, in addition to model number and serial number. Transfer of ownership of those devices will be maintained in the global registry that can reflect all transfers since device manufacturing date as it was described in Section III.

Interested parties, such as operators, distributors, IP protection agencies or even consumers can query and verify a phone through a frond-end interface that uses the main attributes to verify the digital identity. Since the digital identity is cryptographically-based then it can be used to enable tighter authentication and verification means for devices joining a mobile network.

1. Details of Smart Contract Implementation

The use case was implemented using Ethereum as the blockchain technology with the Ethereum Testnet as the network to host the smart contracts. The system is composed of the backend system represented by the public (Testnet) Ethereum network and two smart contracts. One is responsible on maintaining the entities registrar with functions related to defining the registrants and authorizing access to the registry for creating and updating the digital identities. The second contract (the global ID registry) handles the functions of identity creation, identity transfer and identity verification.

For the entities registrar contract, the assumptions is made that an operating entity, in this case the GSM Association, and assignees (e.g. IP protection agency or a regulator) will handle the smart contract management; in terms of adding a registrant or authorizing features. The global registry contract contains a data structure that holds all records of registrants (vendors) and the status. Once more, the registrant is identified by a global and unique identifier based on the Ethereum address and all transactions either for the registrar(s) or registrants are mapped to its blockchain address and secured through the (PK,SK) pair.   Below is a pseudo-code of the entity registrar smart contract.

```
contract globalregistry
{
        address   of main_registrar;
        struct holding
        {
                String of assignee details;
                Flag of assignee privileges;
        }
        struct holding
        {
                Address of registrant;
                String of registrant details;
                Flag of registrant status;
                Flag of registrant privileges;
                Integer holding reputation of registrant;
        }
        Constructor
        {
                Assign main_registrar to address of transaction sender
        }
        // Contract modifiers
        Policy control        modifiers
        {
                // check the identity of the transaction sender
                Is main_registrar;
                Is assignee;
                Is registrant;
                // Check assignee privileges
                Has privileges for adding, updating, deleting, querying a registrant;
                // check registrant enough Ether credit for adding device identities
                Is sender.credit > transaction.value
        }
        // Contract Functions
        Function add_assignee, delete_assignee, update_assignee authorized only to main_registrant
        Function add_registrant, delete_registrant, update_registrant authorized to main_regsitrant or authorized_assignee
} // end of contract
```

Table 1 – Pseduocode of Entity Registrar Smart Contract

For the global IDs registry contract, the contract will hold all the created device identities which will map each smartphone IMEI, blockchain address and other attributes into a unique digital identity. The contract contains a data structure as per Figure 9, where the owner bloackchain address is the registrant address, and the identity flag maps one of four features (updatable, transferrable, withdraw-able, assignable) The contract offers functions related to creating, updating, transferring, verifying, reporting and withdrawing a device identity.

```
contract global_ID
{
        // Device Digital Identity Data Structure
        struct holding
        {
                address of owner;
                address of device;
                bytes20 of attributes-hashedvalue;
                Integer of 1st_block;
                Integer of last_block;
                Integer of ID_revision;
                Flag of identity_features;
                Flag of blocked_device;
        }
        // Array of data structure to hold a linked list of each Identity blocks and revisions
        struct holding
        {
                Integer ID_revision;
                Integer of previous_blockNo;
        }
        Constructor
        {
                Assign main_registrant to address of transaction sender
        }
        // Contract modifiers
        modifiers
        {
                // check the identity of the transaction sender
                Is device_owner;
                Does Identity_Exist;
                // Check registrant privileges
                Has privileges for adding, updating or deleting a device_identity;
                // check registrant enough Ether credit for adding device identities
                Is sender.credit > transaction.value
                Is device_blocked;
        }
        // Contract Functions
        Function create_deviceIdentity;
        Function update_deviceIdentity;
        Function assign_device;
        Function withdraw_deviceIdentity;
        Function query_deviceIdentity_owner;
        Function verifiy_deviceIdentity;
        Function block_device;
} // end of contract
```

Table 2 – Pseduocode of Entity Registrar Smart Contract

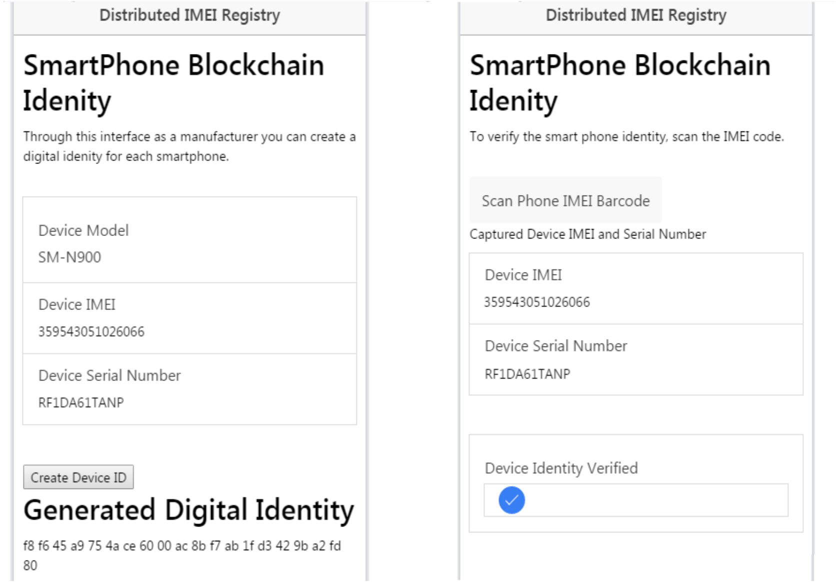
To provide access control few modifiers are used to control identity creation and management; these include confirming device owner identity, verifying registrant privileges, avoiding duplicate identities creation, ensuring registrant’s credit is sufficient. The last modifier will use the Ether crypto-currency or any agreed upon private tokens to enable a charging mechanism for device identity global registry functionality.

2. Use Case Operation

To illustrate how the use case can be operated we list here two screen shots of the front-end application on both the registrant and user side. The screenshots show how first a Registrant creates the smart phone blockchain identity by providing its main attributes (Device Model, IMEI and serial number). The registrant through the interface can transfer the device to a third party (distributor or operator), which as well can transfer the device ownership to the user upon purchase. The front-end through interaction with the smart contract ensure that all these transactions are authenticated using the blockchain address of each entity.

The other screenshot shows how the user or IP protection agency for example through scanning the smartphone barcode can send the hashed value to the smart contract to validate its identity and query its origin and ownership. The validation and verification step is based completely on the digital identity of the device which was created and maintained by the owners.

Finally, a consumer who owns a smartphone and in case of a stolen phone can easily without any 3rd party involvement report its loss and populate that into the global registry of devices blacklist. This through the integration with resellers, operators, law enforcement and insurance will make the fact available by the time the next block is mined in the blockchain (on average 15 sec in the public network). In the centralized case, the user has to report a stolen phone to the local operator, upon that and conditioned on having the operator registered with the GSMA IMEI database, the fact publishing in the central database will take few days to be captured and made available to other parties.



Distributed IMEI Registry

SmartPhone Blockchain Idenity

Through this interface as a manufacturer you can create a digital idenity for each smartphone.

Device Model
SM-N900

Device IMEI
359543051026066

Device Serial Number
RF1DA61TANP

Generated Digital Identity

f8 f6 45 a9 75 4a ce 60 00 ac 8b f7 ab 1f d3 42 9b a2 fd 80

Distributed IMEI Registry

SmartPhone Blockchain Idenity

To verify the smart phone identity, scan the IMEI code.

Captured Device IMEI and Serial Number

Device IMEI
359543051026066

Device Serial Number
RF1DA61TANP

Device Identity Verified
✓

Figure 10 – Front-end for Use Case (Manufacturer-left and Consumer-right)

VII. Conclusion

In this work, we have presented a semi-decentralized approach for identity management in IoT, we introduced a blockchain-based digital identity and identity management solution that ensures a global and unique device identity. The approach introduces the framework and process flow for identity creation and transfer of ownership. Utilizing the underlying blockchain functions and features, specifically cryptographic assets, immutability and provenance, and the proposed structure of the device identity the solution enables identity validation and traceability. The work presents the concepts of BOYDID and highlights how it can enable identity portability which can ensure secure devices onboarding and transfer from one network to another. The proposed approach defines an identity management framework through the device lifecycle.

We also demonstrate that in a use case of mobile phone counterfeiting, the Ethereum blockchain and smart contract provided a tool that can be added to enhance regulations in combating an industry loss of billions of dollars. The use case defines how a smartphone from the day of being manufactured can be assigned a global and unique identity preserved through its lifetime and provides a mean for different stakeholders to validate that identity and report its theft.

It is clear how Blockchains provide a robust distributed system that provides a cryprographic underlying infrastructure for auditable and verifiable identity management. However, we did not discuss issues related to scalability and transactions processing throughput. These issues are topics for continuous research in the blockchain technology that will bring the technology further for adoption in enterprises applications.

References

[1] Vermesan, Ovidiu, et al. "Internet of things strategic research roadmap." *Internet of Things-Global Technological and Societal Trends* 1 (2011).

[2] Trnka, Michal, and Tomas Cerny. "Identity management of devices in Internet of Things environment." *IT Convergence and Security (ICITCS), 2016 6th International Conference on*. IEEE, 2016.

[3] Recordon, David, and Drummond Reed. "OpenID 2.0: a platform for user-centric identity management." *Proceedings of the second ACM workshop on Digital identity management*. ACM, 2006.

[4] Identity Relationship Management 18. (n.d.). Retrieved March 17, 2017, from https://kantarainitiative.org/irmpillars/

[5] Seitz, L., Ed., Gerdes, S., Ed., Selander, G., Mani, M., and S. Kumar, "Use Cases for Authentication and Authorization in Constrained Environments", RFC 7744, DOI 10.17487/RFC7744, January 2016, <http://www.rfc-editor.org/info/rfc7744>.

[6] Chen, Ju, Yi Liu, and Yueting Chai. "An Identity Management Framework for Internet of Things." e-Business Engineering (ICEBE), 2015 IEEE 12th International Conference on. IEEE, 2015.

[7] Empowering the edge. Practical insights on a decentralized Internet of Things. Tech. IBM Institute for Business Value, Apr. 2015. Web. Jan. 2017. <https://www-935.ibm.com/services/multimedia/GBE03662USEN.pdf>.

[8] "Linking the Physical World to the Blockchain." Chronicled. N.p., n.d. Web. Jan. 2017.

[9] “Ethereum's white paper.” Ethereum Foundation, 2014. Retrieved Jan, 2017, from https://github.com/ethereum/wiki/wiki/White-Paper, 2014.

[10] “Ethereum: A secure decentralised generalised transaction ledger.” Gavin Wood, 2014. Retrieved Jan, 2017, from http://gavwood.com/paper.pdf.

[11] “Information technology security techniques: a framework for identity management – part 1: Terminology and concepts,” International Standard ISO/IEC 24760-1:2011.

[12] Bertoni, Guido, et al. "The keccak sha-3 submission." Submission to NIST (Round 3) 6.7, 2011.

[13] IoT Security Foundation. (n.d.). Retrieved March 17, 2017, from https://iotsecurityfoundation.org/

[14] "The Foundation of Internet Identity." OpenID | The Internet Identity Layer. N.p., n.d. Web. Mar. 2017.

[15] "OAuth.net." OAuth 2.0 — OAuth. N.p., n.d. Web. Feb. 2017.

[16] Wajsman, Nathan, and Carolina Arias Burgos. The Economic Impact of Intellectual Property Rights (IPR) Infringement in Smartphone Sector. Rep. European Union Intellectual Property office, Feb. 2017. Web. Mar. 2017. <https://euipo.europa.eu/tunnel.../ip_infringement/.../smartphone_sector_en.pdf>.