Internet of Things Device Capability Profiling Using Blockchain

Internet of Things Device Capability Profiling Using Blockchain

Oluwashina Joseph Ajayi
British Telecom Ireland Innovation Centre
School of Computing
Ulster University
Jordanstown, Northern Ireland, BT37 0QB UK
ajayi-o2@ulster.ac.uk

Joseph Rafferty
British Telecom Ireland Innovation Centre
School of Computing
Ulster University
Jordanstown, Northern Ireland, BT37 0QB UK
j.rafferty@ulster.ac.uk

Philip Morrow
British Telecom Ireland Innovation Centre
School of Computing
Ulster University
Jordanstown, Northern Ireland, BT37 0QB UK
pj.morrow@ulster.ac.uk

Mamun Abu-Tair
British Telecom Ireland Innovation Centre
School of Computing
Ulster University
Jordanstown, Northern Ireland, BT37 0QB UK
m.abu-tair@ulster.ac.uk

Gery Ducatel
British Telecom
Adastral Park
Martlesham, Ipswich, UK
gery.ducatel@bt.com

Zhan Cui
British Telecom
Adastral Park
Martlesham, Ipswich, UK
zhan.cui@bt.com

***Abstract*— Currently, Internet of Things (IoT) devices can integrate into an existing network where they may interact with a myriad of other devices that may host a range of capabilities. Such IoT devices may need to share data that is consumed by other devices or services. This data is generated by the capabilities built into devices within the ecosystem. A typical IoT ecosystem that is heterogeneous in nature should be able to have devices that offer a range of capabilities that could be explored in the event a device breakdown or malfunction. This is to ensure that the system is self-sustaining, and adequately perform during undesirable conditions. Hence, an IoT ecosystem should be able to collaborate, self-organize itself to explore these capabilities towards achieving an overall goal. As such, interoperability of these devices which will improve functionality, availability, and robustness of the IoT ecosystem must be achieved. Also, Several IoT representations today store their data centrally which gives rise to inherent issues such as single point of failure, and other possible vulnerabilities. Addressing these deficiencies alongside proper profiling of IoT device capability and other device details is viewed as the first stage in securing IoT ecosystems and this was explored in this research. This study presents the use of Distributed Ledger Technology which has the inherent property of being secure to profile the capability of IoT devices within self-organized IoT ecosystems. A system overview, data structures, and algorithms are presented.**

Keywords – Autonomous Systems, Blockchain, Device capability profiling, Distributed Ledger Technology, Internet of things, Interoperability, Security, Self-organization.

I. INTRODUCTION

The Internet of Things (IoT) has been defined as "the Future Internet which can be seen as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'things' have identities, physical attribute, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network" [1]. This definition identified some important issues which include the following: (i): "self-configuring capabilities", (ii): "interoperability of protocol", and (iii) "interaction with the existing system". In a typical IoT network, there may exist several devices, and protocols that will need to communicate to share resources and potentially data, in order to collaborate towards a systematic goal. This collaboration involving different types of devices gives rise to issues and challenges related to privacy, trust, risk, and security within both the existing system and the integrated IoT ecosystem. The ecosystem is exposed to several threats, discussed in sub-section A. These threats, if exploited, could lead to system failure irrespective of the overall goal of the system. There are many approaches to realizing networks of IoT devices [2], [3].

The majority of IoT representations are propriety and do not cater for interoperability, may vary in functionality/overall goal, and lack adequate security; among other issues. IoT devices interact with one other, end-users, computational services, and objects to provide functionality. IoT devices are used across a variety of applications, such as healthcare and automotive, and has shown the potential to enhance the capability of services provided [4]. Therefore, devices within an IoT ecosystem need to be properly identified with their capabilities enumerated to enable collaborative functionality, availability, and robustness within the ecosystem.

Device identification and profiling will further support a self-organization paradigm, by allowing devices within the ecosystem to negotiate their capabilities when needed. For instance, if a device with temperature sensing capability is offline and these readings are required by a consuming service, other devices with similar, or closely related, capabilities may be able to provide this data. Such substitute data may be less accurate, or outside of core tolerance, but would still enable the overall solution to operate and will lead to the sustainability of the entire ecosystem. Other related issues are discussed in the subsections below.

A. Threats and Vulnerability in IoT Ecosystem

IoT has enabled the deployment of many applications and has been seen a shift to a pervasive paradigm offering ubiquitous operation from a personal computing paradigm [5]. This adoption of pervasive computing has enabled a range of IoT applications which include smart homes, smart cities, and

978-1-7281-2530-5/19/$31.00 ©2019 IEEE

smart transportation. These applications have resulted in effective resource utilization, reduced human effort and enables organizations to save time in completing activities; thereby increasing productivity [6].

Despite all the benefits of IoT, it has some deficiencies which are briefly discussed here.

Open Web Application Security Project (OWASP) [7], identified major vulnerabilities relevant to an IoT ecosystem, such as insecure network services and interfaces, presence of insecure or outdated components, insecure data transfer, and storage, lack of device management, and insecure default settings.

The identified vulnerabilities make IoT devices subject to malicious threats and attacks such as Distributed Denial of Service (DDoS) and Man in the Middle data interception/manipulation [8] [9]. These threats form an underlining factor for each device to be properly identified and secured while they exist in heterogeneous networks.

B. Centralized IoT Ecosystem

Currently, the most commonly adopted data storage paradigm in IoT systems is centralized in nature. This central deployment can be a myriad of computing paradigms such as Cloud, and Fog. These are discussed in brief below and the associated challenges which call for distributed IoT representation are also discussed.

- *Cloud Computing*

Cloud computing is a paradigm [10] that involves a network of multiple systems connecting to an infrastructure which has the capacity to scale over a period of time. This is usually facilitated through data centers which communicate to consumers via the internet. Cloud computing services typically exist across several models which include: Infrastructure-as-a-Service provides resources such as storage or networking, Platform-as-a-Service provides software-dependent resources for developing, deploying, and managing applications and Software-as-a-Service provides services to end-users and applications [11][12].

- *Fog Computing*

Fog Computing extends/transfers aspects of Cloud Computing outside of the cloud data center, closer to the consuming applications/devices/services. The benefits of this include mobility, low latency transactions, and location awareness, a large number of nodes, disperse geographical distribution, and heterogeneity [13]. These varying characteristics make Fog more appropriate for critical IoT applications such as Wireless Sensors and Actuators Networks (WSANs), Connected Vehicles, and Smart cities [14] where the devices within their ecosystem need immediate response The major aim of Fog computing is to decrease Cloud involvement/reliance by locally filtering/reducing the data generated by IoT devices.

- *Challenges in Cloud, and Fog Computing in relation to IoT*

It can be seen from the above that Cloud servers are at a distance from the devices. A study in [12] summarized challenges or issues with cloud computing as follows: server availability, multitenant services (wherein many subscribers share the same resources), access control (how these services are connected to), the user has no control over the location of storage of their data, data retention (no control over data), and identity protection (this can be seen in relation to privacy). There are benefits associated with using these paradigms as a medium of storing or processing data some of which are on-demand resource allocation - providing scalability, reduced management by end-users, flexibility on pricing and availability of service provisioning/application mechanisms but with this comes some challenges.

Notably, in addition to those mentioned above are three areas described below:

(i) Single point of failure [15]**.** Attackers and adversaries can aim to target data storage as a priority, and because these approaches store, manage, and process a large amount of data centrally, any vulnerability can negatively affect the entire system apart.

(ii) Data/Information Security and Privacy [16] [17]. With computation pushed to the edge of the network in the case of fog computing, information becomes vulnerable to various security attacks. Trust management schemes, such as automatic knowledge monitoring [18], human-based trust protocol [19], and green trust management [20], which can also enforce security is also missing. User data can be visible to the public, which leads to the threats of privacy invasion.

(iii) Data Ownership [21]. Currently, there are no mechanisms to prevent highly private data to either be removed from the cloud/fog/ before processing or storage. For example, in mobile applications, the data collected by IoT devices will be stored and analyzed by the service provider.

(iv) Deficient Architecture and Susceptibility to Manipulation [15] [22] [23].The likelihood of information manipulation and inappropriate use can arise because of the block of IoT architectures in these paradigms which acts as a bottleneck and can disrupt communication across the entire network.

The identified deficiencies have given rise to looking for an alternative way to represent IoT deployment. This, therefore, suggests a distributed representation. In a distributed network, any successful attack will have less impact as resources are distributed over the network [23]. An attack on the ecosystem will have to be doubled to bring the network down because, with an attack on one entity, the other entities can have a copy of the original information.

C. Distributed IoT Ecosystem Representation

Establishing a secure paradigm within IoT networks requires a distributed communication platform which guarantees security, transparency, and immutability [24].

A distributed and decentralized approach which provides replica data across nodes and is immutable in nature would resolve many challenges with a centralized approach such as found in the cloud, or fog [25]. Additionally, it has been noted that blockchain can provide high levels of security for IoT devices [25]. Distributed Ledger Technology (DLT) may provide the foundation for devices to interact without having any central control whilst simultaneously maintaining individual privacy, ensuring trust, and providing security [26].

Several studies have explored the use of blockchain for the possibility of securing IoT systems [24] [27] [28]. These studies have shown that it is important to also establish interoperability among networks that are incompatible [26] in various ways such as protocols and varying architectures. Achieving this will enable the realization of a distributed representation of an IoT

ecosystem where different devices and protocols co-exist and communicate with each other without any central control. DLT is distributed in nature and can provide trusted data without a centralized server and may be applied within the foundation of a solution for self-organizing, secured, IoT networks [29].

A self-organizing system will typically contain several devices that interact with each other and their surroundings. Individual devices usually exhibit a simple behavior which when combined with other autonomous devices within the network results in a complex system with improved capabilities [30]. For instance, devices with close capability can co-exist in an IoT network, one of them may be redundant and be ready to provide alternative capabilities should an active device become unable to fulfill its role. This will increase the availability of the IoT solution. The device capability profiling based on blockchain presented in this research will, therefore, lay a good foundation for a secure self-organized IoT ecosystem.

The remainder of this paper is structured as follows; Section II explores related work. Section III presents our approach, describing the system overview based on blockchain, the proposed algorithm, and the data structure that enables assessment of the capabilities of trusted devices within the ecosystem. Conclusions and future work are presented in Section IV to extend the functionality of this platform.

II. RELATED WORK

There have been several studies on maintaining resilient interoperability of IoT devices to form resilient solution. These have been based on many technologies including semantic ontologies [31] and software agents [32].

Park et al. [31] proposed a technological platform which provides semantic-based IoT information services and semantic interoperability of IoT devices. The service was aimed toward smart devices gathering information on an environment, sharing this information to participating devices and consuming services. Their methodology was developed out of concern that IoT applications need to be aware of heterogenous IoT middleware platforms, sensors, and networks to function. Another concern was the adoption of proprietary Application Programming Interfaces (APIs) which makes it difficult for applications and devices to access various resources and services which are associated with the middleware.

In cases where an application can interact with the middleware, they must search, collect, analyze and process the data themselves. Their work did not, however, identify the nature of the device in terms of functionalities, their protocols and did not detail how platform agnostic middleware can be achieved.

Goumopoulous and Kemeas [33] proposed a component-oriented programming model alongside a middleware designed towards smart objects. Their concepts focused on artifacts which have objects with properties, compositions, and synapses which shows the association between the artifacts. The interfaces shared by the smart objects and the rules governing them are determined by the middleware. Their work did not address how the properties of artifacts and compositions can determine how heterogeneous devices can communicate.

A context-aware multi-layered agent for smart objects was proposed by Fortino et al. [34]. The approach presented here is basically based on a software agent paradigm. A framework that provides a heterogeneous middleware approach based on Semantic Web was presented by Song et al. [35]. A semantic layer which mapped each device to Web Ontology Language for Services (OWL-S) based Semantic Web service was proposed [35]. They provided an example that shows how to add semantic information to devices using Universal Plug and Play (UPnP) and Bluetooth. Details on how other communication protocols can be integrated were however not provided in their study.

IoT hubs were used to aggregate devices with services using web protocols in a study by Blackstock and Lea [36]. The architecture proposed in this study consists of the following: IoT Core (where things and their metadata are exposed as RESTful web services); IoT Model (this involves the development of adapters and other integration tools); IoT Hub (that provides agreement on implementation issues); IoT Profiles (provide agreement on the semantics of things and their associated data exposed on a hub). They later concluded that the new challenge lies in the interoperability or unification of hub catalogs and data formats as there are several IoT hubs that aggregate IoT resources.

Proposed by Desai et al. [37] is a solution which incorporates an architecture involving a gateway and Semantic Web-enabled IoT architecture that provides interoperability using communication and data standards. In their study, translation between protocols such as Extensible Messaging and Presence Protocol (XMPP), Constrained Application Protocol (CoAP) and Message Queuing Telemetry Transport (MQTT) commonly used in IoT communication was enabled using a multi-protocol proxy. A concept of Semantic Gateway as Service (SGS) was established to provide a bridge between devices and services they interact with. The devices in the topology connect to the gateways using CoAP, XMPP, or MQTT protocol while interacting with the Semantic Gateway which provides the translation. They cited some key standardization efforts such as Open Geospatial Consortium Sensor Web Enablement (OGC SWE) which is a standard model and XML Schema for observations and sensors. Despite the mentioned standardization effort, Desai et al. [37] claim that the interoperability challenges on IoT have many unmet requirements and a semantic IoT architecture or other alternatives are required to support multiple IoT protocols.

Xiao et al. [38] proposed User Interoperability Framework to solve the interoperability problem between an IoT device user of a context and a device of another context using consistent syntax and semantics. This involved using different IoT devices with different capabilities and functionality. In their study, heterogeneous devices are separated into different device classes, namely a real device, a common device and a virtual device, where any device is both syntactically and semantically transformable within a different context. They noted that there is a high-level problem in communication between devices and users in heterogeneous settings. Other studies based on a semantic data model are by Boyi Xu et al. [39], Nambi et al. [40].

A related study on sensitive feature profiling of IoT devices by [32] provides a framework that profiles security and privacy-sensitive functionality of a device without having physical access to them. This framework is based on semantic analysis of discovered technical information of the devices such as types of communication protocol, transmission range, and type of

sensing and actuating equipment. This software-based framework that automatically detects device profiles was applied to a fitness tracker, and a voice-controlled smart home assistant.

In current IoT service platforms representation, devices and applications need to have knowledge of how to access middleware which is a limitation in IoT ecosystems [31]. The goal is to develop an open IoT service platform that can provide devices with the following services: seamless access to and use of IoT resources and data (interoperability); a secure and trusted connection between devices and resources [31]. This will result in varying platforms, capabilities and different communication technologies. This then requires the development of an IoT platform that requires devices to be able to automatically join and integrate with existing infrastructure which is the vision of IoT [41]. Sematic ontology discussed by various authors is deficient in its reusability [42]. Hence, this study will explore software agent which is seen as a supporting technology to drive interoperability among heterogeneous solutions [43]. In achieving this, this paper intends to create a novel device capability profiling based on blockchain and to support interoperability of devices. The following are the major contributions of this study:

Contribution 1: To create a distributed representation of devices capabilities (physical attributes, communication protocols and enumerations of data provided/consumed) on a blockchain to enable safe communication between devices.

Contribution 2: To attempt to create interoperability among different communication and messaging protocols that exist in a range of devices. This will be achieved by using a software-based agent.

Contribution 3: To enhance security and create an auto-discovery service for heterogeneous devices of different capabilities (constrained and non-constrained) – *Algorithms for implementation is proposed*.

III. OUR APPROACH

We present a novel mechanism for device capability profiling, discovery and goal resolution using a foundation incorporating distributed ledger technology. This approach attempts to provide a solution to the deficiencies of centralized IoT ecosystems. Proprietary gateway devices are capable of automated device registration and deregistration but due to the heterogeneous nature of IoT, a platform that can handle these devices is required [31]. IoT ecosystems are highly heterogeneous and to maximize its potential, interoperability must be achieved [44].

Our approach looks at a broad representation of device capability with respect to the following; physical capabilities, communication protocol, data messaging protocol, and data format (details provided in Table 1). To achieve this, a permissioned blockchain called Hyperledger Fabric which provides a first level of security to IoT devices during profiling is used as a distributed platform, this is discussed in subsection A below. A permissioned blockchain is required to properly identify the peers within the network and to enforce security.

A. Hyperledger Fabric

Hyperledger Fabric [45] is an established and active framework by Linux foundation which provides a blockchain architecture which aims to provide resiliency, flexibility, scalability, confidentiality, and security to any entity associated with it [46]. Fabric became the first distributed platform for permissioned blockchain due to the ability to execute standard programming languages consistently across all nodes [45]. This is achieved by the introduction of "System Chaincode" which may be used for transaction processing, configuration and also assists with maintenance of the distributed network [47].

Hyperledger Fabric incorporates the concept of Smart Contracts which enable the performance of credible transactions which are traceable, and irreversible.

The smart contract that runs on Fabric peers and creates transactions that logically decide how ledgers are stored in the world state (database) is called "System Chaincode". Fig. 1 shows an annotated diagram of Fabric and its modular architecture. Fabric provides a flexible approach for implementation because of its modular representation.

TABLE 1
DEVICE DETAILS DATA STRUCTURE

Physical feature	Device Name Manufacturer Serial Number Firmware version Device location	Data Format String/Plain Text	*TDS JSON
Protocol Type	Communication: WiFi, 6LowPAN ZigBee, Z-Wave Bluetooth	String/Plain Text	JSON
	Data Messaging: MQTT, CoAP XMPP, AMQP	String/Plain Text	JSON
Transport	TCP, UDP	String/Plain Text	JSON

***Translated Data Structure**

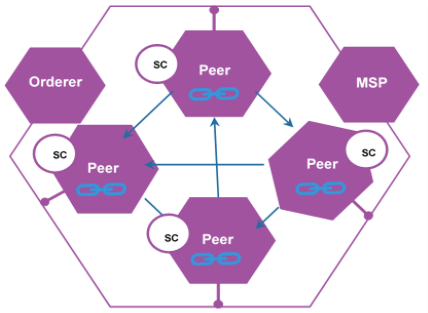

Fig. 1: Hyperledger Modular representation interaction

From fig. 1, Fabric achieves its modular representation based on its "orderer", peers, System Chaincode (SC), Membership Service Provider (MSP), and Ledger. Entities in Figure 1 are now briefly explained. The order of transactions is established through consensus by the orderer (ordering service) which atomically broadcasts state updates to peers. The MSP associate's peers with cryptographic identities and helps to maintain the permissioned nature of Fabric.

A peer-to-peer service disseminates the output of the block through the ordering service to all peers. The system chaincode

runs in a container environment, such as Docker, and can be programmed using variety of programming languages such as JavaScript [47]. Finally, ledgers are locally maintained on the peers. IoT devices can be represented as a peer or node on Fabric. There are use cases of Fabric and more than 400 prototypes, proofs-of-concept, and in production across different industries; this is due to its maturity, and continuing support by Linux foundation. Use cases include but are not limited to food safety, identity management, contract management, trade logistics and dispute resolution [45].

B. Overall System Overview

Fig. 2 shows the system overview of the proposed solution. It consists of seven steps which are briefly explained below.

In step 1, as identified in the previous section, the device capabilities include the physical capabilities, communication, and data messaging protocols, and device data format. The cluster of devices using its communication protocol through the communication layer associate itself with the network provided with the aid of the "discovery service". The discovery service will also be inbuilt with an interoperable capability by either implementing a semantic ontology or a software agent that seamlessly handshake with the device at this level.

Devices are associated with a corresponding transport layer protocol in step 2 with the aid of the discovery service which, by means of Representational State Transfer (REST) in Step 3, translate to the Hypertext Transfer Protocol (HTTP) that connects to the Internet stack. Security at this level can be maintained by using Transport Layer Security for Transmission Control Protocol, and Datagram Transport Layer Security for User Datagram Protocol. Using REST and HTTP, the Fabric Software Development Kit (SDK)/API or Command Line Interface connects to the blockchain in step 4.

The discovery service is represented on the chaincode of the blockchain to synchronize the system state/status.

In step 5, the membership service provider or a chosen certificate authority generates an identity for the device and associated capability as determined by the discovery service.

The identity in the form of a key and certificate is passed on to the discovery service and to the device through the communication layer in step 6. The device key and certificate (holding its identity and capability) can be verified on the blockchain in step 7 via the communication layer. Tables 1 and 2 detail the device capability and desired functionality data structure. The possible data format and the translated data format are described and expanded in section C below.

C. Capability Assessment Algorithms

Table 1 presents the device details data structure while table 2 presents the desired functionality data structure used in this study. The tables show the data structure of each requirement and the required translated data format to be used. The data structures presented in the tables are used closely in our proposed algorithms. For clarity, the proposed algorithms were broken down into three major segments (Algorithm A, B, and C) major segments of the "Overall Capability Assessment".

TABLE 2
DESIRED FUNCTIONALITY DATA STRUCTURE

Functionality	Variant	Data Format	TDS
Device Type	Sensors Actuator Tag	String/Plain Text	JSON
Sensing Type	Ex. Humidity Temperature	String/Plain Text	JSON
Blockchain Anchor	Certificate/Key Digital Signature	Plain Text to Cypher	Keys

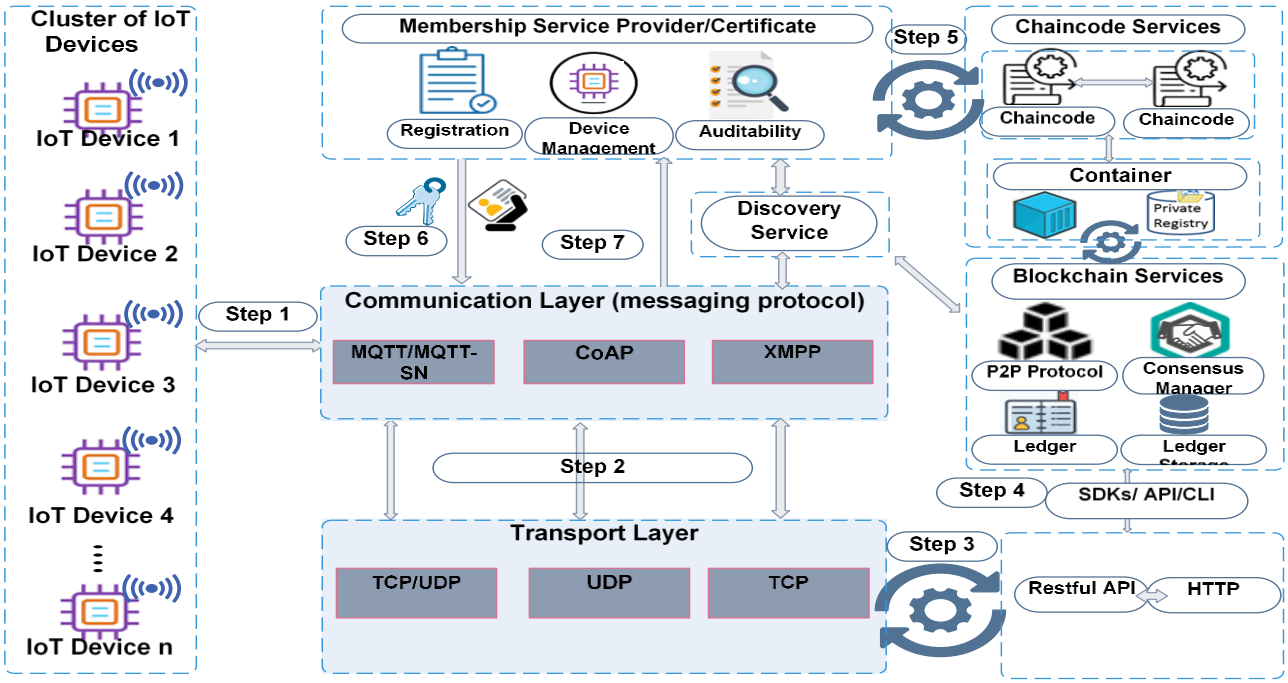

Fig. 2: Overall System Overview

In algorithm A, range of devices (constrained, and unconstrained) are identified as input. The expected output is as follows: device details (*deviceDetails*), and device data structure (*deviceDataStructure*). A function called "check_details" is defined. This function detects device details as detailed in Table 1. The device data structure and device details are returned as output. This function terminates when the total number of devices within the network has been exhausted. The device details are returned as JSON and used by "Algorithm B" as input.

Algorithm A: Algorithm for Acquisition of Device Details

```
Input: device d
Output: deviceDetails, deviceDataStructure
Function: check_details // to acquire device details
1       {detect device details}
2               deviceName: dataformat
3               manufacturer: dataformat
4               deviceSerialNo: dataformat
5               firmwareVersion: dataformat
6               deviceLocation: dataformat
7               communicationProtocol: dataformat
8               dataMessagingProtocol: dataformat
9               transportLayerProtocol: dataformat
10              return deviceDataStructure,
11                      deviceDetails
12              //translated data structure as shown in table 1
13              // this is returned as JSON
14      // This device details are used as an input for
15      // generating its capability in Algorithm B
```

Algorithm B receives "deviceDetails", "deviceDataStructure", as input from Algorithm A, alongside

Algorithm B: Algorithm for Generating Device Capability

```
Input: deviceDetails, deviceDataStructure //from Algorithm A
       Device d, deviceType, deviceSensingType
Output: deviceCapability
Function: deviceFunctionality //parameter passed:
          deviceDetails, deviceDataStructure
1     {detect deviceCapability}
2       deviceFunctionality:deviceDetails, deviceDataStructure
3         //for device details, and data structure, refer to Table 1
4         //and 2
5       deviceType:dataformat
6       deviceSensingType:dataformat
7          return deviceCapability
8         //translated data structure as shown in table 2
9         // this is returned as JSON
```

"deviceType", and "deviceSensingType". The output here is the "deviceCapability" which is achieved by the function called "deviceFunctionality". Algorithm B translate the output of Algorithm B to provide the capabilities embedded in the devices as JSON format. This uniquely translates the capability of each device.

The "deviceCapability" from Algorithm B is used in Algorithm C – "Algorithm for Blockchain connection". The output for this algorithm is a "deviceKey" which is a combination of a key and certificate that uniquely identify a device. This is achieved using a function called "connecToBlockchain" which accept the "deviceCapability" as JSON and connect to the MSP of Hyperledger Fabric. The assigned key can be used by the devices during interaction for authentication, and authorization.

Notably, in the Overall Capability Assessment Algorithm below are, the *discovery service*, *agent service*, and *blockchain service* (as also shown in fig. 2 – *"Overall System Overview"*). The discovery service to be implemented will be enhanced by the "*ZeroConf*" also known as "*bonjour"* protocol which has four major requirements; Internet Protocol (IP) interface configuration, translation between the hostname and IP address, IP multicast address allocation, and service discovery. The service discovery will enable the discovery of the Agent services and blockchain service. This will tightly interface between the device, agent services, and the blockchain service. This algorithm called the executable functions from Algorithm A, B, and C respectively.

The *Agent Service* which will be connected through an Application Programming Interface (API) to the discovery service will be implemented to perform the following functions: (i) to enhance interoperability of different protocols (communication and data messaging – refer to Table 1), (ii) to provide a translation of different data types (identified in table 1) to JavaScript Object Notation (JSON). This then connects to the *blockchain service* through a client Software Development Kit (SDK) via the transport protocol Transmission Control Protocol (TCP/IP) or User Datagram Protocol (UDP) depending on the Agent service translation.

On the *Blockchain service*, the *membership service provider* of Hyperledger Fabric will be used to provide keys to the devices. These keys will be used by the devices for

Algorithm C: Algorithm for Blockchain connection

```
Input: deviceCapability //from Algorithm B
       device d
Output: deviceKey
Function: connecToBlockchain //this is enabled by parsing the
          //deviceCapability in JSON to the Membership Service
          //Provider (MSP) of Hyperledger Fabric via an API to
          //the SDK
1     request certificate and signature for d //through
2     //the discovery service to the membership service
3     //provider of Hyperledger Fabric for key
4     //assignment
5     connecToBlockchain (deviceCapability)
6           generate deviceKey //based on Capability
7       return deviceKey
8   assign deviceKey to d
```

communication among themselves. The key will also be tied closely with the device capability and this will be visible to the discovery services as well (overall description is given in section B).

For the remainder of the algorithm, after these processes have been established, for a range of devices, the device details will be checked via the *discovery service*, and the data structure will be returned via the *Agent service*.

Algorithm: Overall Capability Assessment Algorithm

Input: device d
Output: deviceDetails, deviceDataStructure, deviceCapability, deviceKey
1 ***initialize*** local network ***and start*** discovery service
2 ***initialize*** blockchain service
3 ***connect*** device d to local network
4 ***initialize Agent*** for protocols interoperability support
5 device d ***connect*** through *discovery service*
6 device d *consume* the ***Agent service*** through ***discovery***
7 for device d, i = 1 to n //*where n is the number of devices*
8 request certificate and signature for d //*through*
9 //*the discovery service to the membership service*
10 //*provider of Hyperledger Fabric for key*
11 //*assignment*
12 ***Call:***
13 **function** check_details (d) *//from Algorithm A*
14 *//dataFormat now translated to JSON via Agent Service*
15 **function** deviceFunctionality (deviceDetails,
16 deviceDataStructure) *//Algorithm B*
17 **function** connecToBlockchain (d) *//Algorithm C*
18 **endCall**
19 endfor
20 ***device d*** ready to interact using ***deviceKey***
21 end

This is returned in JSON format. The device will be profiled based on the device details, and its translated data structure and returned as the device capability. The identified device capability will be passed on to the membership service provider of the *blockchain service* which generates the device key and passed on to the device. The device will use this key for authentication and authorization while communicating within the network. This will provide the first level of security. The device will also be able to announce their capability at an interval for a possible interaction with other devices.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel approach for profiling device capabilities based on Hyperledger Fabric. Our approach is to create a distributed representation of device capabilities via blockchain transactions in order to create an enabling secure communication between devices and perform the basis of an autonomous system which enables dynamic collaborative solutions. To this end, a *system overview was presented.* Interoperability which is very important to our proposal will be achieved either through the implementation of a software agent on the discovery service which will have the capability to translate protocols, and data format. Also provided are possible *algorithms* for device capability assessment. We believe that if an IoT device capability is well identified and associated with blockchain, it can provide the first level of security between heterogeneous devices (constrained and non-constrained) in an IoT Ecosystem. Since this is ongoing research, the next step will be to implement the proposed concept as part of an overall approach in managing diversity within varying IoT devices with different capabilities. An interoperability approach will be finalized and implemented on the discovery service which will then interact with the blockchain. Computationally constrained and unconstrained IoT devices will be used for the implementation. This will form a testbed that will be used for the proof of concept. Once this is completed, the solution will be tested using propriety IoT devices.

REFERENCES

[1] H. Sundmaeker, P. Guillemin, P. Friess, and Sylvie Woelfflé, *Vision and Challenges for Realising the Internet of Things*, vol. 1, no. March. 2010.

[2] Y. Chen, "Challenges & Oppoetunities in IoT," *IEEE Conf. Wirel. Sensors*, vol. 16, no. 12, pp. 383–388, 2012.

[3] R. Gunasagaran *et al.*, "Internet of things: Sensor to sensor communication," *2015 IEEE SENSORS - Proc.*, pp. 1–4, 2015.

[4] B. Al-Shargabi and O. Sabri, "Internet of Things: An exploration study of opportunities and challenges," *Proc. - 2017 Int. Conf. Eng. MIS, ICEMIS 2017*, vol. 2018-Janua, pp. 1–4, 2018.

[5] A. Radovici, C. Rusu, and R. Serban, "A Survey of IoT Security Threats and Solutions," *Proc. - 17th RoEduNet IEEE Int. Conf. Netw. Educ. Res. RoEduNet 2018*, pp. 1–5, 2018.

[6] N. Cam-Winget, A. R. Sadeghi, and ..., "Can iot be secured: Emerging challenges in connecting the unconnected," *ACM J. Des. Autom.*, 2016.

[7] OWASP, "OWASP Top 10 Internet of Things," *Salem Press Encycl. Sci.*, pp. 5–7, 2018.

[8] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A Security Taxonomy for IoT," *Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018*, pp. 163–168, 2018.

[9] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things : Architecture , Enabling Technologies , Security and Privacy , and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, 2017.

[10] H. Wadhwa and R. Aron, "Fog Computing with the Integration of Internet of Things: Architecture, Applications and Future Directions," *2018 IEEE Intl Conf Parallel Distrib. Process. with Appl. Ubiquitous Comput. Commun. Big Data Cloud Comput. Soc. Comput. Networking, Sustain. Comput. Commun.*, pp. 987–994, 2018.

[11] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[12] Z. Tari, X. Yi, U. S. Premarathne, P. Bertok, and I. Khalil, "Security and privacy in cloud computing: Vision, trends, and challenges," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 30–38, 2015.

[13] A. Seal and A. Mukherjee, "On the Emerging Coexistence of Edge, Fog and Cloud Computing paradigms in Real-Time Internets-of-EveryThings which operate in the Big-Squared Data space," *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2018-April, pp. 1–9, 2018.

[14] M. Saad, "Fog Computing and Its Role in the Internet of Things: Concept, Security and Privacy Issues," *Int. J. Comput. Appl.*, vol. 180, no. 32, pp. 7–9, 2018.

[15] R. Roman, J. Zhou, and J. Lopez, "On the features and Challenges of Security & Privacy in Distributed Internet of Things," *Comput. Networks*, vol. 57, 2013.

[16] H. El-Sayed *et al.*, "Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2017.

[17] M. Ashouri, P. Davidsson, and R. Spalazzese, "Cloud, edge, or both? Towards decision support for designing IoT applications," *2018 5th Int. Conf. Internet Things Syst. Manag. Secur. IoTSMS 2018*, pp. 155–162, 2018.

[18] Z. Movahedi, M. Nogueira, and G. Pujolle, "An autonomic knowledge monitoring scheme for trust management on mobile ad hoc networks," *IEEE Wirel. Commun. Netw. Conf. WCNC*, pp. 1898–1903, 2012.

[19] P. B. Velloso, R. P. Laufer, D. D. O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "Trust management in mobile Ad Hoc networks using a scalable maturity-based model," *IEEE Trans. Netw. Serv. Manag.*, vol. 7, no. 3, pp. 172–185, 2010.

[20] Z. Hosseini and Z. Movahedi, "A Green Trust Management Scheme to Mitigate Trust-Distortion Attacks on MANETs," *Proc. - 13th*

IEEE Int. Conf. Ubiquitous Intell. Comput. 13th IEEE Int. Conf. Adv. Trust. Comput. 16th IEEE Int. Conf. Scalable Comput. Commun. IEEE Int. Conf. Cloud Big Data Comput. IEEE Int. Conf. Internet People IEEE Smart World Congr. Work. UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld 2016, pp. 518–525, 2017.

[21] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[22] N. Kshetri, "Can Blockchain Strengthen the Internet of Things?," *Secur. IT*, no. August, pp. 68–72, 2017.

[23] H. Y. Shwe, T. K. Jet, P. Han, J. Chong, and A. S. Architecture, "An IoT-oriented data storage framework in smart city applications - IEEE Xplore Document," *2016 Int. Conf. Inf. Commun. Technol. Converg.*, pp. 106–108, 2016.

[24] P. Siano, G. De Marco, A. Rolan, and V. Loia, "A Survey and Evaluation of the Potentials of Distributed Ledger Technology for Peer-to-Peer Transactive Energy Exchanges in Local Energy Markets," *IEEE Syst. J.*, pp. 1–13, 2019.

[25] N. Kshetri, "Can Blockchain Strengthen the Internet of Things?," *Secur. IT, a Publ. IEEE Comput. Soc.*, no. August, pp. 68–72, 2017.

[26] D. Burkhardt, M. Werling, and H. Lasi, "Distributed Ledger: Definition & Demarcation," *2018 IEEE Int. Conf. Eng. Technol. Innov. ICE/ITMC 2018 - Proc.*, pp. 1–9, 2018.

[27] P. Ferraro, C. King, and R. Shorten, "Distributed Ledger Technology, Cyber-Physical Systems, and Social Compliance," *IEEE Access*, vol. 6, pp. 1–19, 2018.

[28] W. Gao, W. G. Hatcher, and W. Yu, "A Survey of Blockchain : Techniques , Applications , and Challenges," *2018 27th Int. Conf. Comput. Commun. Networks*, no. i, pp. 1–11, 2018.

[29] R. Kuhn, D. Yaga, and J. Voas, "Rethinking Distributed Ledger Technology," *Computer (Long. Beach. Calif).*, vol. 52, no. 2, pp. 68–72, 2019.

[30] A. Sobe, I. Fehérvári, and W. Elmenreich, "FREVO: A tool for evolving and evaluating self-organizing systems," *Proc. - 2012 IEEE 6th Int. Conf. Self-Adaptive Self-Organizing Syst. Work. SASOW 2012*, pp. 105–110, 2012.

[31] D. H. Park, H. C. Bang, C. S. Pyo, and S. J. Kang, "Semantic open IoT service platform technology," *2014 IEEE World Forum Internet Things, WF-IoT 2014*, pp. 85–88, 2014.

[32] A. Bytes, S. Adepu, and J. Zhou, "Towards Semantic Sensitive Feature Profiling of IoT Devices," *IEEE Internet Things J.*, vol. PP, no. c, pp. 1–1, 2019.

[33] C. Goumopoulos and A. Kameas, "Smart objects as components of UbiComp applications," *Int. J. Multimed. Ubiquitous Eng.*, vol. 4, no. 3, pp. 1–20, 2009.

[34] G. Fortino, A. Guerrieri, and W. Russo, "Agent-oriented smart objects development," *Proc. 2012 IEEE 16th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2012*, pp. 907–912, 2012.

[35] Z. Song, A. A. Cárdenas, and R. Masuoka, "Semantic middleware for the internet of things," *2010 Internet Things, IoT 2010*, pp. 1–8, 2010.

[36] M. Blackstock and R. Lea, "IoT interoperability: A hub-based approach," *2014 Int. Conf. Internet Things, IOT 2014*, pp. 79–84, 2014.

[37] P. Desai, A. Sheth, and P. Anantharam, "Semantic Gateway as a Service Architecture for IoT Interoperability," *Proc. - 2015 IEEE 3rd Int. Conf. Mob. Serv. MS 2015*, pp. 313–319, 2015.

[38] G. Xiao, J. Guo, L. Da Xu, and Z. Gong, "User interoperability with heterogeneous IoT devices through transformation," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1486–1496, 2014.

[39] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in iot-based information system for emergency medical services," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1578–1586, 2014.

[40] S. N. A. U. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim, "A unified semantic knowledge base for IoT," *2014 IEEE World Forum Internet Things, WF-IoT 2014*, pp. 575–580, 2014.

[41] E. Chindenga, C. Gurajena, and M. Thinyane, "Towards an adaptive ontology based model for interoperability in internet of things (IoT)," *2016 IST-Africa Conf. IST-Africa 2016*, pp. 1–8, 2016.

[42] M. Ma, P. Wang, and C. H. Chu, "Ontology-based semantic modeling and evaluation for internet of things applications," *Proc. - 2014 IEEE Int. Conf. Internet Things, iThings 2014, 2014 IEEE Int. Conf. Green Comput. Commun. GreenCom 2014 2014 IEEE Int. Conf. Cyber-Physical-Social Comput. CPS 2014*, no. iThings, pp. 24–30, 2014.

[43] N. Khalid, H. F. Ahmad, and H. Suguri, "Software agents mediated interoperability among heterogeneous semantic services," *Proc. - 2008 IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. IAT 2008*, vol. 2, pp. 144–147, 2008.

[44] A. Botta *et al.*, "On the Integration of Cloud Computing and Internet of Things," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 23–30, 2013.

[45] E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *Proc. EuroSys 2018 Conf. Distrib. Parallel, Clust. Comput.*, pp. 1–15, 2018.

[46] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," *2017 IEEE Technol. Eng. Manag. Soc. Conf. TEMSCON 2017*, no. 2016, pp. 137–141, 2017.

[47] T. Sato, Y. Himura, and J. Nemoto, "Design and Evaluation of Smart-Contract-based System Operations for Permissioned Blockchain-based Systems *," *IEEE J. Networks*, 2018.