Device identity management on Hyperledger Fabric

Ananya Tandon
School of Computing and Augmented Intelligence, Arizona State University
Tempe, USA
atando16@asu.edu

Anil Kumar
School of Computing and Augmented Intelligence, Arizona State University
Tempe, USA
akuma254@asu.edu

Dragan Boscovic
School of Computing and Augmented Intelligence, Arizona State University
Tempe, USA
dragan.boscovic@asu.edu

***Abstract*—With a surge in demand for IoT products and most IoT end markets, IoT Analytics anticipates the chip shortage to have a long-term impact on the number of connected IoT devices. It is predicted that there will be around 27 billion linked IoT devices by 2025 [9], as supply restrictions ease and demand increases. The majority of these devices are deployed in diverse and complex networks, posing several problems to the device management functions. One of these problems is identity management, which is concerned with how devices' identities are reconciled and confirmed, as well as how devices construct the methods for authorizing and managing access to their data and services. Blockchain, being a distributed ledger technology, promotes itself as a viable solution to the problem of Identity management. This is mostly due to Blockchain providing record immutability, cryptographical IDs, and data provenance. These capabilities, when combined, provide a foundation for implementing IoT device identity management services and a means to preserve it over the device life cycle. Our work describes a decentralized Hyperledger fabric-based IoT identity management platform that includes features such as data collection from the manufacturer itself, ownership transfer, identity generation, and detecting device anomalous operational state.**

***Index Terms*—Blockchain, Hyperledger fabric, Identity Management, Internet of Things, Smart Contracts.**

I. Introduction

In the Internet of Things (IoT), identity management plays a growing role in identifying, securing, and controlling devices' access to various types of ecosystems. It includes addressing the issue of providing a unique identity to devices, controlling how they interact with one another and the services they provide, and restricting access to sensitive data and services. A crucial development necessitates that Identity and Access Management capabilities be able to manage interactions between different devices, among devices and users, and between devices and services.

IoT systems frequently have sporadic or temporary connections and before the interaction can start, the endpoints must execute identification, authentication, and authorization in compliance with a well-managed identity management life cycle that can provide a secure and trusted autonomous environment. As IoT devices gain popularity in both the consumer and business arenas, there are also growing numbers of devices and diverse interactions among them. Centralized data management systems are being used by the majority of existing systems where a single authority holds the data ownership which leads to problems, particularly in situations where devices must communicate data in a peer-to-peer method. The decentralized approach over the problems of data immutability, shared ownership of resources, decentralized data provenance management, and IoT security [3]. Based on its immutability characteristic and the anonymity of its identification method, the Hyperledger fabric is a viable solution for the creation of decentralized IoT platforms that can preserve a safe and trusted data exchange, and anomaly detection in real-time.

We have used Hyperledger Fabric to store the device data continuously during the device's lifespan and generate the unique device IDs using the dynamic state parameters using smart contracts. We have trained machine learning models in parallel using device historical data to detect anomalous device behavior in real-time. The strategy also offers a way to enable shared ownership, transfer device ownership, and update the device.

The following document has been structured into different sections. Section II puts limelight on the related work and section III elaborates the nature of data and all the dynamic state parameters being collected over the life span of the IoT device. section IV will elaborate on the data storage on Hyperledger fabric-based blockchain infrastructure involving multiple network stakeholders, its implementation, device ID creation, how it operates and how multiple stakeholders agree upon a common device state by using smart different contracts. In section V, we describe the details of the proposed solution and anomaly detection in the device's operational states. Finally, section VI covers the summary and the future impact of the work.

II. Related Work

While initial proposals [8] for a blockchain-based IoT identity management system appear in the literature, they focus primarily on practicality and not functionality. Identity management frameworks are heavily constrained by the need to ensure the capabilities of authentications, auditing, and permissions governance across the identity management life cycle [10]. To comply with these restrictions, unambiguous registration procedures must be established, global and unique identities must be maintained, and frameworks for access control and authorization must be provided. Numerous research has been undertaken to improve data security in light of these issues. The article presents a blockchain-based authentication

method [7]. Each participant may be authenticated by validating the left adjoining member and other approaches [4] provide a smart contract-based approach with striking similarities to our design, but their stakeholder model and lack of anomaly detection limit its applicability in practice."

The study [1] discusses group sensor communication methods as a technique for addressing numerous concerns in wireless data transmission. For sensor node authorization and data transfer, this system executes fundamental arithmetic and logical functions. [5] provides more comprehensive review of the existing work in this area.

III. Data Collection

A digital twin is a representation of a part of a physical equipment or component in the actual world [6]. It connects the physical and digital worlds via streams of data produced by a particular device. We maintain a digital twin of an IoT device that contains its network parameters, network packet information, and data from its sensors. This information is captured in a JSON format and sent to Hyperledger Fabric (HLF) every 30 seconds while the device is running which is stored as a dynamic time series on HLF.

A. Use Case and Device

We tested our implementation on a Solar Panel cleaning the aerial ground vehicle that removes dust, sand, or leaves with the pointed wind at regular intervals and prevents the accumulation of soiling in arid regions. The device is based on Raspberry Pi and can be accessed remotely through Secure Shell(SSH).

1) Specifications:

- Hardware platform: Raspberry Pi 3b; Architecture: 32 bit
- Operating system: Raspberry Pi OS; Type: Debian

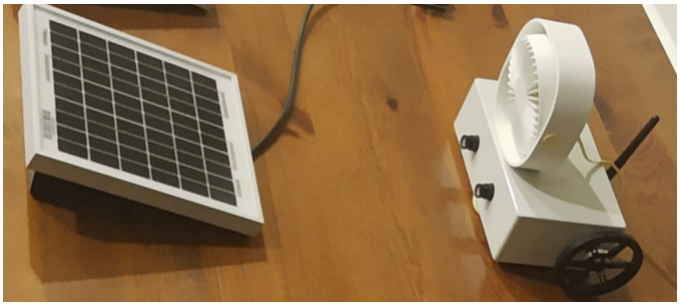

Fig. 1. Solar Panel Cleaning Aerial Ground Vehicle

B. Parameters Captured

1) Static Parameters: The following static parameters that remain the same throughout the life cycle of the device are captured to establish the identity of the digital twin on HLF. The static parameters captured are Device Name, Owner, Status (Active/Inactive), Mac Address, Serial Number, Manufacturer, Hardware, RAM, and OS Information.

Note: Parameters are not captured on HLF when the device is inactive

2) Dynamic Parameters: Dynamic parameters keep changing according to the mode of operation of the device, these parameters form the digital twin of the device. All the historical data consisting of these dynamic parameters are also used for training a Random Forest Classifier which can detect the operational state of the device in real-time. The following dynamic parameters can be captured from any raspberry pi based device using the same code:

- IP Addressing: Device, Broadcast and Gateway IP, Netmask, and Domain Name System(DNS) information.
- Packet Information: DNS state, Transmission Control Protocol(TCP)connections, Transport Layer Security(TLS) state, Internet Control Message Protocol(ICMP) Type/Error
- Kernel Process Information: Tasks, Memory utilization, and Process-specific information.
- System Activity Report: Input/Output Statistics, CPU Utilization, Disk Device Status, Processor Queue Information
- Wireless Access Points visible to the device
- IP and MAC Addresses of all Devices in the network

However, some dynamic features should be captured according to the specific use case of the device. These features identify the actual mode of operation of the device, these are used to create labels for training the random forest classifier.

- Status of USB Ports
- Values from the Sensors(Encoders and Proximity Sensors)
- Status of General Purpose Input Output(GPIO) Pins of the raspberry pi.

IV. Data management over Hyperledger Fabric

We'll talk about managing Internet of Things (IoT) devices using Hyperledger fabric in this part.

A. Hyperledger Fabric Network Initialization

Hyperledger Fabric is an open-source, and enterprise-ready private distributed ledger technology. It has advanced privacy protections to guarantee that only the data you want to be shared is shared with the network's "permitted" (known) actors. The decentralized blockchain network is used to distribute the code and the agreements it contains. Businesses are more confident since transactions are traceable and final. This reduces time, cost, and risk by enabling firms to make smarter decisions more rapidly. The suggested method offers a solitary platform for customers to register their devices, transfer ownership of the devices, and gain permitted access to device life cycle data. A security operator may also utilize this platform to create machine learning models and track state changes over time. Therefore, the major players who have access to device data are device owners and security operators. Owners can let others join communication channels and have access to device life cycle data.

We now outline the steps involved in constructing the network for managing digital twins:

1) We have three different organizations in the Hyperledger fabric network named Manufacturer, Owner, and Security Analyst.
2) All three network stakeholders are being hosted over three different Google cloud platform instances, having one peer node each. The manufacturer creates the digital twin of the device over the Blockchain network. The manufacturer can allow the Owner organization to join the network channel to pass the ownership to the new party.
3) While creating the digital twin, every device gets a unique ID from the blockchain network. Once the device gets the unique ID, it's allowed to record its dynamic state parameters over the blockchain network.
4) Security Analysts are allowed over the network channel by the owner to have access to the device data. Security Entity continuously accesses the data of the device and creates the machine learning model using the historical data.
5) Machine learning model is integrated with updated device smart contract over the blockchain network and verifies the device's operational state in real-time.
6) Whenever a security entity founds a discrepancy in the device state, it monitors the device operational state for the next few device states and informs the device owners if the state parameters pattern doesn't change.

B. *Smart contracts*

To create interactions that are added to the ledger, applications call a smart contract. Once the corresponding contract is executed from a valid account, Hyperledger Fabric nodes obey the instructions in the SC code. For our experiment, we have three different smart contracts in the HLF network. One smart contract is introduced to register new users and updated the existing users over the network. The second smart contract is responsible for the ownership transfer of the devices. The third smart contract is used to update the digital twin of IoT devices over the network. The fourth smart contract is mainly used by security operators to access the device state parameters and integrate the machine learning models with the blockchain network. We have integrated the public key on all of the contracts so that if a malicious entity tries to manipulate the device state, the blockchain network simply ignores them. We now elaborate the motive of all the smart contracts:

1) User Management: Manufacturers are default stakeholders of the network and initially they can invite other parties to join the network over communication channels. New users can use this smart contract to join a network and update the stakeholder information.
2) Sell/Purchase Device: Device ownership will vary over its lifetime. This smart contract was used to transfer the ownership of the IoT device over to the new users.
3) Update Device: Operating IoT devices continuously update the dynamic state parameters reflecting their digital identity using the ledger state. This smart contract was used to log the device state every 30 seconds. Machine learning models were integrated with this smart contract to verify the device's operational state with the required state of the device.
4) State Management: All the device states were monitored over time by using the machine learning model trained over the device's historical data. In case of required changes over the device's operational states due to the new operational scenarios this smart contract was used to install new updates on the device and upgrade existing device-specific machine learning models; for example, an increase in memory consumption due to library updates.

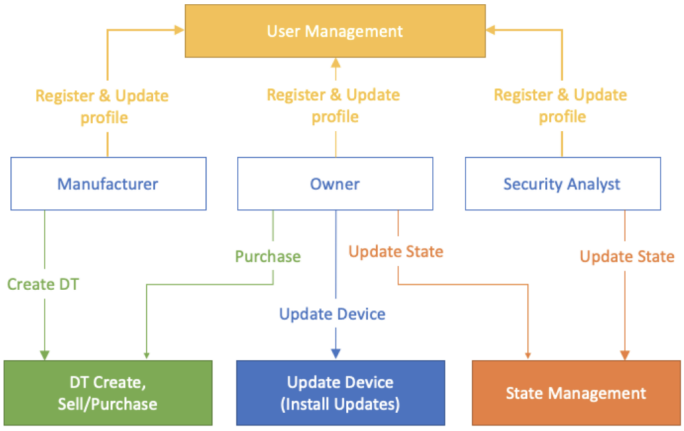

Fig. 2. Smart Contracts Interaction Diagram

Figure 3 represents different stakeholders of the network with their attributes and all the operations performed by smart contract to update the ledger state.

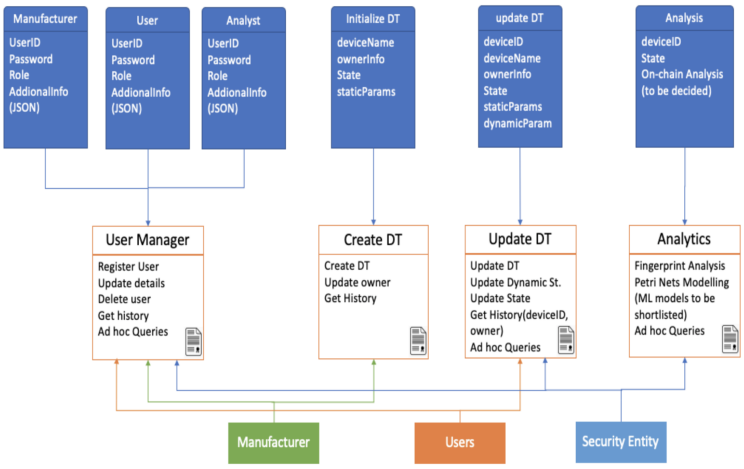

Fig. 3. Actors and relationship diagram with Smart contracts

V. Operational States and Anomaly Detection

A. *State Detection*

The state or mode of operation of the device is detected using a random forest classifier as we are working with high

dimensional data of different types. Random Forest Classifier is robust to outliers and can handle binary, categorical as well as numerical features. Once we have detected the mode of operation of the device, we can deduce its identity from them. The 4 device states, that are used to label the training data were decided based on wheel and fan movements as shown below:

States		Fan	
		On	Off
Wheels	On	**Fan On/ Moving**	**Fan Off/ Moving**
	Off	**Fan On/ Not Moving**	**Idle**

Fig. 4. Operational states of the device

For each use case, the features used for training will be different, these features are a subset of all the parameters. We select these features by looking at the correlation matrix as well as experimentation. From Kernel Processes we get the features: Memory Used, Buffer/Cache Memory, and Swap Memory Available. From the System Activity Report we get the Amount of memory needed for the current workload and an Estimate of how much memory is available for starting new applications, without swapping. These features can be obtained from any raspberry pi based device. Using these features alone the model was able to predict with 72% accuracy. However, when we added distance from the solar panel as a feature, captured using proximity sensors the accuracy of the model increased to 94%. Which was an expected result because the states are highly reliant on the distance from the sensor.

B. Anomaly detection

Any scenario involving anomaly detection requires the definition of ideal operational conditions. It can be decided both in advance and when the IoT device is regularly using it. To train machine learning models, the system must determine the device's operational states. The operational stages for our use case were based on fan and wheel movements, however, they can change depending on the type of IoT device.

For our use case, we were collecting device dynamic state data every 30 seconds. Machine learning models trained over the historical data were integrated with smart contracts of Hyperledger fabric-based backend and on receiving the incoming data from the device smart contract compares the input given to the robot (obtained from GPIO pins) with the sensor information. If these values do not match, it becomes evident that the device is showing anomalous behavior. For example, if the GPIO pins are getting the signal to move the wheel but the encoder senses no rotation of wheels, it could mean that either the wheels are not getting enough voltage for movement or the internal wiring is compromised.

VI. Conclusion

For our use case, we were successfully able to detect the device's operational state with 94 percent of accuracy and were able to perform anomalous state detection which in our case was the device operating far away from solar panels. The use case reflects the potential of blockchain in IoT device security along with shared ownership of the device data. Although privacy and dependability issues with IoT could be resolved by using blockchain technology [2]. This technique does, nevertheless, have significant drawbacks that make it difficult. The ledger storage limitations, limited technological advancements, a skilled labor shortage, a lack of appropriate legal rules and standards, variances in processing times and speeds, computer power limitations, and scalability constraints are a few of these difficulties. Further research will investigate whether Training data can be captured in a sliding window manner to accommodate continuous updates on the device. We can incorporate device updates in real-time, for which we will need to detect new Ideal working states in real-time.

References

[1] H. Kim and J. Kang. "Dynamic Group Management Scheme for Sustainable and Secure Information Sensing in IoT". In: *Sustainability* 08 (2016).

[2] Cristian Martín Ana Reyna. "On blockchain and its integration with IoT. Challenges and opportunities". In: *Future Generation Computer Systems* 88 (2018), pp. 173–190.

[3] Khaled Salah Minhaj Ahmad Khan. "IoT security: Review, blockchain solutions, and open challenges". In: *Future Generation Computer Systems* 82 (2018), pp. 395–411.

[4] Otman Basir Ahmad Sghaier Omar. "Identity Management in IoT Networks Using Blockchain and Smart Contracts". In: *Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (2019).

[5] Youakim Badr Xiaoyang Zhu. "A Survey on Blockchain-Based Identity Management Systems for the Internet of Things". In: *Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (2019).

[6] Chris Snider David Jones. "Characterising the Digital Twin: A systematic literature review". In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52.

[7] F Li Z Xu. "A blockchain-based authentication and dynamic group key agreement protocol". In: *Sensors* 20 (2020).

[8] Sahraoui Dhelim Mohammed Amine Bouras Qinghua Lu. "A Lightweight Blockchain-Based IoT Identity Management Approach". In: *Future Internet* 13 (2021).

[9] Mohammad Hasan. *State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally*. URL: https://iot-analytics.com/number-connected-iot-devices/.

[10] Richard Hill Martin Kuppinger. *Identity Governance Administration(IGA)*. URL: https://www.rsa.com/wp-content/uploads/KuppingerCole-Identity-Governance-and-Administration.pdf.