Journal of Software Engineering and Applications, 2016, 9, 533-546
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Blockchain Platform for Industrial Internet of Things

Arshdeep Bahga, Vijay K. Madisetti

Georgia Institute of Technology, Atlanta, GA, USA
Email: arshdeepbahga@gmail.com, vkm@gatech.edu

How to cite this paper: Bahga, A. and Madisetti, V.K. (2016) Blockchain Platform for Industrial Internet of Things. *Journal of Software Engineering and Applications*, **9**, 533-546.
http://dx.doi.org/10.4236/jsea.2016.910036

Received: August 15, 2016
Accepted: October 25, 2016
Published: October 28, 2016

Abstract

Internet of Things (IoT) are being adopted for industrial and manufacturing applications such as manufacturing automation, remote machine diagnostics, prognostic health management of industrial machines and supply chain management. Cloud-Based Manufacturing is a recent on-demand model of manufacturing that is leveraging IoT technologies. While Cloud-Based Manufacturing enables on-demand access to manufacturing resources, a trusted intermediary is required for transactions between the users who wish to avail manufacturing services. We present a decentralized, peer-to-peer platform called BPIIoT for Industrial Internet of Things based on the Block chain technology. With the use of Blockchain technology, the BPIIoT platform enables peers in a decentralized, trustless, peer-to-peer network to interact with each other without the need for a trusted intermediary.

Keywords

Internet of Things, Blockchain, Smart Contracts, Cloud-Based Manufacturing

1. Introduction

Internet of Things (IoT) comprises “Things” (or IoT devices) which have remote sensing and/or actuating capabilities, and can exchange data with other connected devices and applications (directly or indirectly). IoT devices can collect data and process the data either locally or send the data to centralized servers or cloud-based application back-ends for processing [1]. IoT technologies are promising for industrial and manufacturing systems. Experts have forecast a trillion dollar impact of IoT on the industrial and manufacturing sectors. A recent on-demand model of manufacturing that is leveraging IoT technologies is called Cloud-Based Manufacturing (CBM) [2]. CBM enables ubiquitous, convenient, on-demand network access to a shared pool of configurable manufacturing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [3] [4].

In this paper, we propose a decentralized, peer-to-peer platform called BPIIoT for Industrial Internet of Things based on Blockchain, the technology that powers the crypto-currency Bitcoin. The BPIIoT platform will act as a key-enabler for cloud-based manufacturing, enhancing the functionality of existing CBM platforms, especially towards integrating legacy shop floor equipment into the cloud environment. While Cloud-Based Manufacturing enables on-demand access to manufacturing resources, a trusted intermediary is required for transactions between the users who wish to avail manufacturing services. With the use of Block chain technology, the BPIIoT platform enables peers in a decentralized, trustless, peer-to-peer network to interact with each other without the need for a trusted intermediary.

2. Related Work

The Blockchain technology was introduced along with Bitcoin by Satoshi Nakamoto [5]. The Bitcoin transactions are recorded in a public ledger called the Blockchain. IBM and Samsung have announced a collaboration to build decentralized IoT solutions by leveraging the Blockchain technology [6].

Slock it has developed a smart lock technology called Slocks which enables real-world physical objects to be controlled by the Blockchain [7]. The owners of a Slock who wants to rent their real-world physical objects (such as houses, cars or bikes) set a deposit amount and a price for using the objects. Users can find the Slocks using the mobile app and then make a payment in Ethers to rent the objects. After the transactions are validated on the Ethereum Blockchain network, the users get permission to open or close the Slocks with their smartphone. A smart contract is automatically enforced between the owner and the user. After the object is returned, the deposit minus the cost of the rental is returned to the user.

Trans Active Grid has developed a combination of software and hardware technologies that enable users to buy and sell solar energy from each other securely and automatically, using smart contracts and the Blockchain [8]. Filament has built an open technology stack based on Blockchain technology, to enable devices to discover, communicate, and interact with each other in a fully autonomous and distributed manner [9]. A review of Blockchain and smart contracts and applications of Blockchain for IoT are described in [10].

3. Key Contributions of Proposed Work

We propose a Blockchain Platform for Industrial Internet of Things (BPIIoT), which enables development of different distributed apps (Dapps) for manufacturing. **Figure 1** shows the difference between an existing solution (Slock.it) and the proposed BPIIoT platform. While existing solutions like Slock.it can be considered as Dapps specialized to perform a specific task (such as unlocking a lock), BPIIoT has a much broader scope as it enables development of different decentralized and peer-to-peer manufacturing applications as described in Section-6. These manufacturing Dapps build upon the

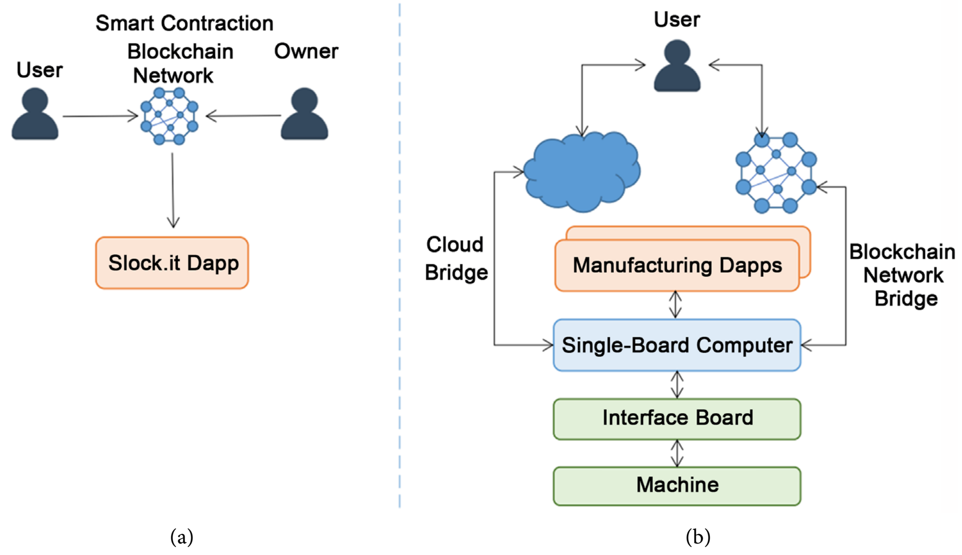


Figure 1. Slock.it vs BPIIoT: (a) Slock.it Dapp is based on a smart contract between a user and the owner of a real-world physical object, to unlock the lock when the user pays a rent and deposit to the smart contract to unlock the lock (b) BPIIoT platform has a much broader scope than Slock.it as it enables development of different decentralized and peer-to-peer manufacturing applications.

BPIIoT platform comprising a single-board computer (with cloud and Blockchain bridges, sensors and actuator connectivity drivers, Blockchain wallet, controller service, I/O and device management components) and an interface board.

BPIIoT also enhances existing CBM platforms by: 1) enabling consumer-to-machine and machine-to-machine transactions without a trusted intermediary; 2) integrating legacy shop floor equipment into the cloud environment; 3) automating machine maintenance and diagnostics tasks; 4) providing a distributed, secure and shared ledger of all transactions, assets and inventory records.

4. Blockchain Concepts

In this section, we describe the key concepts related to Blockchain in general and the Ethereumblock chain network which we have used for our proposed BPIIoT platform.

- **Blockchain**: Blockchain is a distributed data structure comprising a chain of blocks. Blockchain acts as a distributed database or a global ledger which maintains records of all transactions on a Blockchain network. The transactions are time stamped and bundled into blocks where each block is identified by its cryptographic hash. The blocks form a linear sequence where each block references the hash of the previous block, forming a chain of blocks called the Blockchain. A Blockchain is maintained by a network of nodes and every node executes and records the same transactions. The Blockchain is replicated among the nodes in the Blockchain network. Any node in the network can read the transactions. Figure 2(a) shows the structure of a Blockchain.

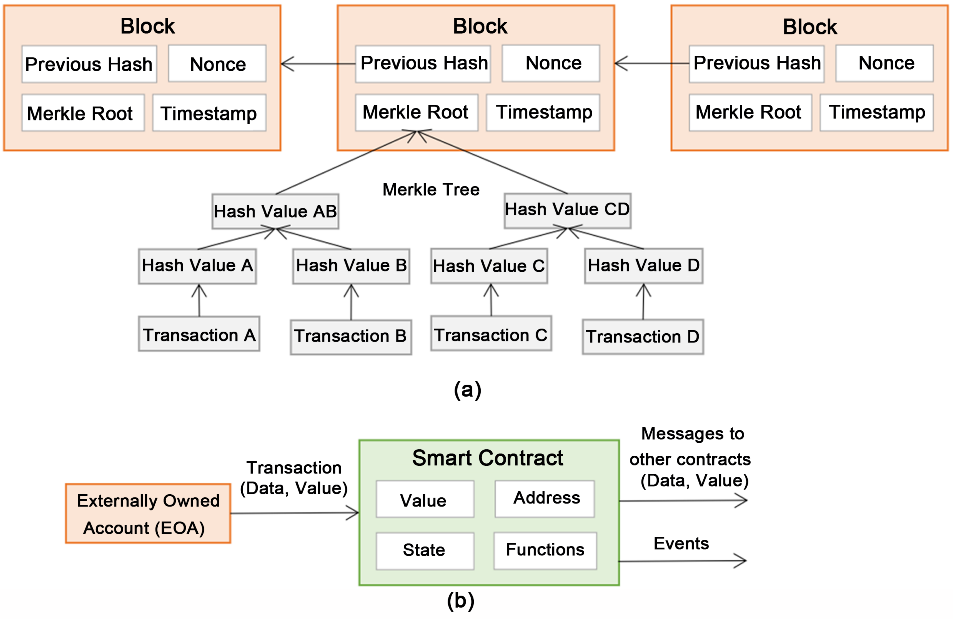


Figure 2. (a) Blockchain structure; (b) Smart contract structure.

- **Smart Contracts**: A smart contract is a piece of code that resides on a Blockchain and is identified by a unique address. A smart contract includes a set of executable functions and state variables. The functions are executed when transactions are made to these functions. The transactions include input parameters which are required by the functions in the contract. Upon the execution of a function, the state variables in the contract change depending on the logic implemented in the function. Contracts can be written in various high-level languages (such as Solidity or Python) [11]. Language-specific compilers for smart contracts (such as Solidity or Serpent) are used to compile the contracts into byte code. Once compiled the contracts are uploaded to the Blockchain network which assigns unique addresses to the contracts. Any user on the Blockchain network can trigger the functions in the contract by sending transactions to the contract. The contract code is executed on each node participating in the network as part of the verification of new blocks. Figure 2(b) shows the structure of a smart contract.
- **Ethereum**: Ethereum is an open and programmable Blockchain platform [12]. Anyone can sign up for the platform and create an Ethereum account. Users can create and deploy smart contracts to the Ethereum platform and build decentralized applications. The platform is not owned or controlled by a single entity and is powered by the peers who run the Ethereum nodes.
- **Ethereum Virtual Machine (EVM)**: Ethereum Virtual Machine (EVM) is the runtime environment for smart contracts in Ethereum. The nodes in the Ethereum network run the EVM. The EVM runs as a sandbox and provides an isolated execution environment. All the nodes in the Blockchain network perform the same computations thus providing redundancy in the execution of smart contracts. While this massive amount of redundancy is not an efficient approach for execution, but it is

required to maintain consensus in the network where there is no centralized authority or a trusted third-party.

- **Ethereum Accounts**: Ethereum has two types of accounts—Externally Owned Accounts (EOAs) and Contract Accounts. EOAs are the accounts which are owned and controlled by the users. Each EOA has an Ether balance associated with it. These accounts can send transactions to other EOAs or contract accounts. The contract accounts are controlled by the associated contract code which is stored with the account. The contract code execution is triggered by transactions sent by EOAs or messages sent by other contracts.
- **Public-Private Keys**: Each Externally Owned Account (EOA) has a public-private key pair associated with it. The account address is derived from the public key. When a new EOA is created, a JSON key file is created which has the public and private keys associated with the account. The private key is encrypted with the password which is provided while creating the account. For sending transactions to other accounts, the private key and the account password are required.
- **Dapp**: A Decentralized Application (or Dapp) is an application that uses smart contracts. Dapps provide a user-friendly interface to smart contracts. A crypto currency application is an example of a Dapp that runs on a Blockchain network.
- **Blocks**: The transactions in a Blockchain network are bundled into blocks and executed on all the participating nodes. A block contains a transaction list, the most recent state, a block number and a difficulty value. If there are conflicting transactions on the network (for example, transactions that do double spending), only one of them is selected to become a part of the block. The blocks are added to the Blockchain at regular intervals.
- **Transactions**: Transactions are the messages which are sent by Externally Owned Accounts (EOAs) to other EOAs or contract accounts. Each transaction includes the address of the recipient, transaction data payload and a transaction value. When a transaction is sent to an EOA, the transaction value is transferred to the recipient. When a transaction is sent to a contract account, the transaction data payload is used to provide input to the contract function to be executed. Transactions are signed by the sender's private key. Transactions are selected and included in the blocks in the mining process. The state of the network is changed only by the transactions which are selected for inclusion in the blocks. The transactions on a Blockchain network can be read by all the participant nodes in the network.
- **Messages**: Contracts deployed on a Blockchain network can send messages to other contracts. A message contains the address of the sender, address of the recipient, value to transfer and a data field which contains the input data to the recipient contract. The difference between a transaction and a message is that a message is produced by a contract while a transaction is produced by an EOA.
- **Mining**: The transactions on a Blockchain network are verified in a process called mining. The participating nodes in the network are given incentives in the form of Ethers for performing the mining operations. Miners compete to do a complex ma-

thematical computation and the node that wins, earns a reward in Ethers. Miners produce blocks which are verified by other miners for validity. A valid block is one which contains proof of work (PoW) of a given difficulty. In Ethereum, a proof-of-work algorithm called Ethash is used. The PoW algorithm finds an *nounce* input to the algorithm so that the result is below a certain difficulty threshold. The time for finding a new block can be controlled by manipulating the difficulty. A successful PoW miner is one whose block is selected to be next on the Blockchain. Once a winning block is selected all other nodes update to that new block.

- **Ether**: Ether is the currency which is used in the Ethereum Blockchain network. The miners in the Ethereum network receive mining rewards in the form of Ethers. The base unit of Ether is called Wei (where 1 Ether = 10^{18} Wei).
- **Gas**: Gas is the name of the crypto fuel which is consumed for performing the operations on a Blockchain network. All the transactions on the network are charged a certain amount of gas. While sending a transaction, the sender sets a gas price which represents the fee the sender is willing to pay for gas. The senders of the transactions are charged a gas fee, which is paid to the miners and the balance is refunded to the sender. The gas fee paid is proportional to the amount of work that is needed to execute the transaction, in terms of the number of atomic instructions.

5. Blockchain Platform for Industrial Internet of Things

We propose a Blockchain Platform for Industrial Internet of Things (BPIIoT). Figure 3 shows how BPIIoT platform can enhance the functionality of Cloud-based Manufacturing (CBM) platforms, by providing a decentralized, trustless, peer-to-peer network for manufacturing applications. CBM is a service-oriented manufacturing model in

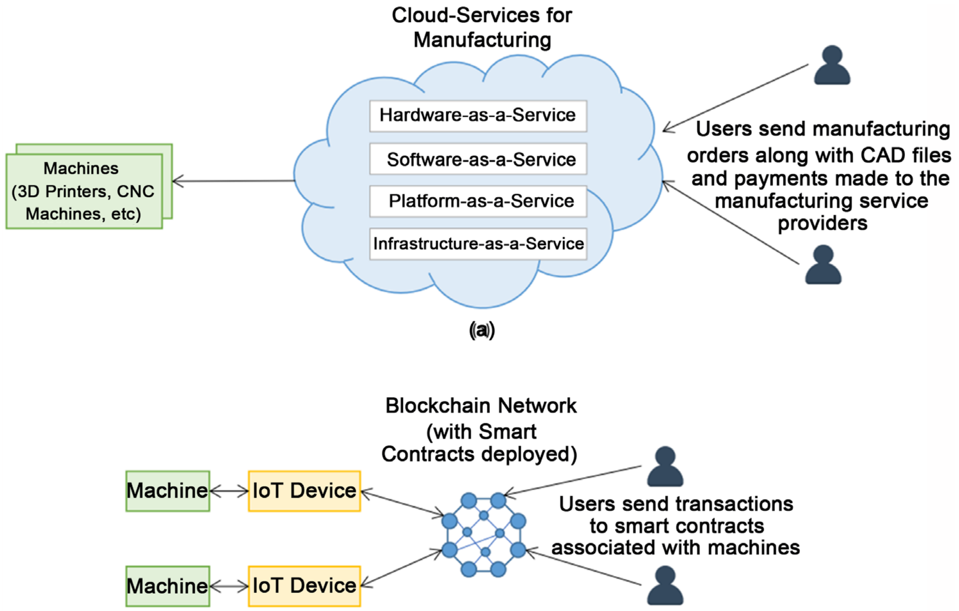

Cloud-Services for Manufacturing
Hardware-as-a-Service
Software-as-a-Service
Platform-as-a-Service
Infrastructure-as-a-Service
Machines (3D Printers, CNC Machines, etc)
Users send manufacturing orders along with CAD files and payments made to the manufacturing service providers
(a)
Blockchain Network (with Smart Contracts deployed)
Machine
IoT Device
Machine
IoT Device
Users send transactions to smart contracts associated with machines

Figure 3. (a) Cloud-based manufacturing (CBM) model; (b) Blockchain for Industrial Internet of Things.

which service consumers are able to configure, select, and utilize configurable manufacturing resources. CBM leverages the four key cloud computing service models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Hardware-as-a-Service (HaaS), and Software-as-a-Service (SaaS) [13]. BPIIoT is based on a Blockchain network on which smart contracts are deployed. The smart contracts act as agreements between the service consumers and the manufacturing resources to provide on-demand manufacturing services. BPIIoT enables integrating legacy shop floor equipment into the cloud environment and allows developing decentralized and peer-to-peer manufacturing applications.

The key enabler component for the industrial machines in the proposed BPIIoT platform is the IoT device. Figure 4 shows the architecture of the IoT device. The IoT device enables existing machines to communicate with the cloud as well as the Blockchainnet work. The IoT device is a "plug and play" solution that allows machines to exchange data on their operations to the cloud, send transactions to the associated smart contracts and receive transactions from the peers on the Blockchain network.

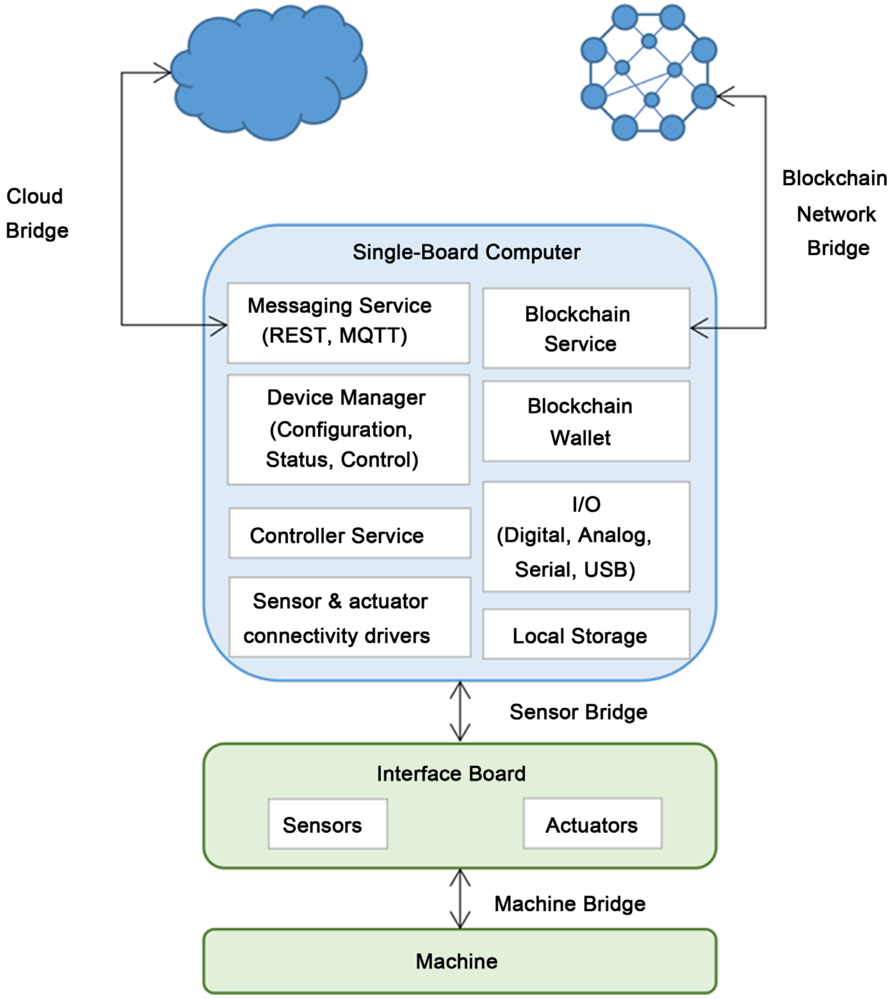

Cloud Bridge
Blockchain Network Bridge
Single-Board Computer
Messaging Service (REST, MQTT)
Blockchain Service
Device Manager (Configuration, Status, Control)
Blockchain Wallet
Controller Service
I/O (Digital, Analog, Serial, USB)
Sensor & actuator connectivity drivers
Local Storage
Sensor Bridge
Interface Board
Sensors
Actuators
Machine Bridge
Machine

Figure 4. Blockchain platform for industrial internet of things (BPIIoT).

The IoT device includes an interface board (based on Arduino) and a single-board computer (based on Beaglebone Black or Raspberry Pi). The interface board has digital input/output and analog input capability. Sensors and actuators interface with the digital or analog pins on the interface board. The interface board has a serial interface to the single-board computer (SBC). The sensor bridge between the interface board and the SBC enables the SBC to capture sensor data from the interface board and also send control signals to the actuators. The sensor and actuator connectivity drivers are installed on the SBC. The device manager on the SBC allows the users to configure the SBC using a web interface and also view the device status and statistics. The I/O block on the SBC enables connectivity to external systems over digital, analog, serial and USB connections. The Blockchain service on the SBC communicates with the Blockchain network and sends/receives transactions to/from the network. Each IoT device has its own account on the Blockchain network and maintains a Blockchain wallet on the SBC. The controller service performs various actions which can be configured through the device management interface. For example, an action can be to capture the data from the sensor bridge and publish it to the cloud through the cloud bridge. Another action can be to monitor the machine status and operating environment, and send transactions to the associated smart contracts on the Blockchain network (such as a transaction to order replacement of a part).

6. Applications of BPIIoT Platform

In this section we describe some decentralized industrial and manufacturing applications that can leverage the proposed BPIIoT Platform.

- **On-Demand Manufacturing**: The BPIIoT platform will enable a marketplace of manufacturing services where the machines will have their own Blockchain accounts and the users will be able to provision and transact with the machines directly to avail manufacturing services in a CBM-like on-demand model. BPIIoT can enable peer-to-peer networked manufacturing applications where the peers will be able to avail manufacturing services (such as CNC machining or 3D printing) by sending transactions to the machines.
- **Smart Diagnostics & Machine Maintenance**: BPIIoT platform can be used for developing smart diagnostics and self-service applications for machines where the machines will be able to monitor their state, diagnose problems, and autonomously place service, consumables replenishment, or part replacement requests to the machine maintenance vendors. Smart contracts between manufacturers and vendors for procurement of supplies and service of machines can help in automating the machine maintenance tasks. We describe the implementation of a smart diagnostics and machine maintenance application using the proposed platform in Section-7.
- **Traceability**: BPIIoT platform can be used for developing traceability applications for manufactured products. Smart contracts between the consumers and manufacturers can keep production records, for example, which factory and which machines within the factory were used for manufacturing a particular product. In case of

product recalls (either due to manufacturing defects or faulty parts) after the products are delivered, traceability applications can help in identifying the affected products.

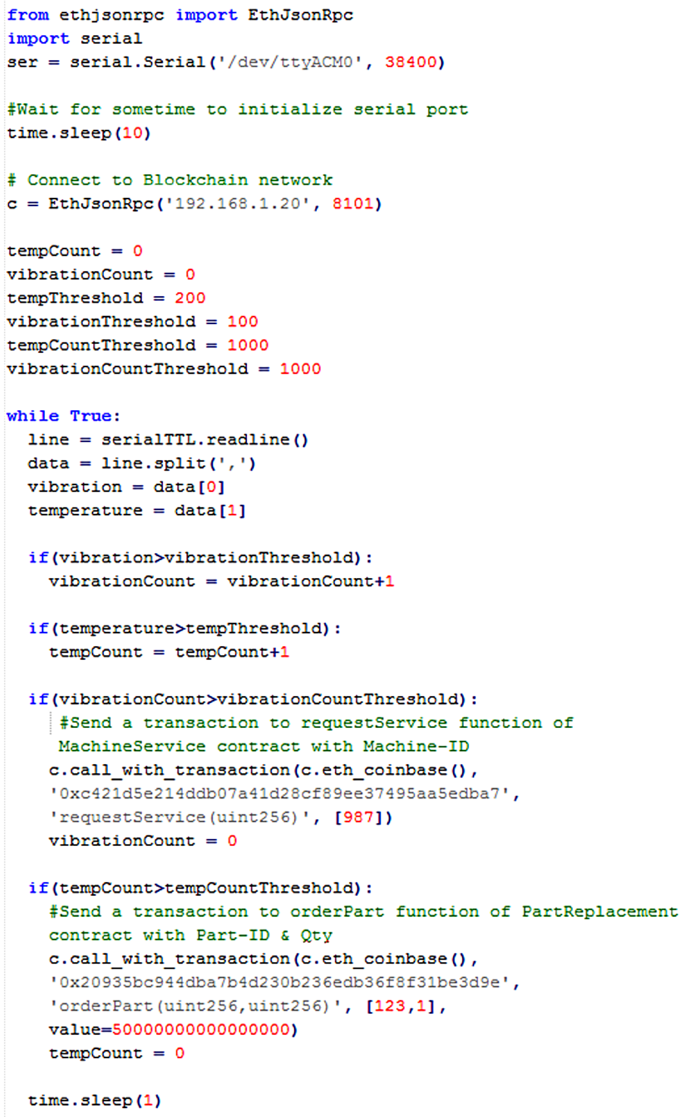
- **Supply Chain Tracking**: BPIIoT platform can be used for developing supply chain tracking applications. The Blockchain and Smart Contracts can keep a formal registry of products and track their possession through different points in a supply chain. Such applications can also enable automated financial settlements on delivery confirmations.
- **Product Certification**: BPIIoT platform can be used for developing product certification applications. The manufacturing information for a product (such as the manufacturing facility details, machine details, manufacturing date and parts information) can be recorded on the Blockchain. This information can help in proving the authenticity of the products eliminating the need for physical certificates which can be prone to tampering and forging.
- **Consumer-to-Machine & Machine-to-Machine Transactions**: BPIIoT can enable machine-to-machine transactions for manufacturing services. For example, a consumer can send a request for manufacturing a product by sending a transaction to a manufacturer's smart contract along with the payment made in a crypto currency (such as Ethers). The manufacturer's smart contract can then send transactions to smart contracts associated with individual machines (consumer-to-machine transactions). If services of different machines are required for manufacturing a product, the machines can send micro-transactions to other machines (machine-to-machine transactions).
- **Tracking Supplier Identity & Reputation**: BPIIoT platform can be used for developing supplier identity and reputation management applications which track various performance parameters (such as delivery times, customer reviews and seller ratings) for sellers. Such applications can be used by consumers to find the sellers that can meet their manufacturing requirements and by manufacturers for finding suppliers for consumables. Smart contracts can also help in automatically negotiating best prices for consumables and supplies in real-time based on the seller reputations.
- **Registry of Assets & Inventory**: BPIIoT platform can be used for developing applications for maintaining records of manufacturing assets and inventory. These applications can keep records of the asset identification information and the transfer of assets through the supply chain eliminating the need for manual paper records.

7. Implementation Case Study

In this section, we describe an implementation case study of the proposed BPIIoT platform based on Beaglebone Black single-board computer and an interface-board based on Arduino Uno. To demonstrate the proposed platform we describe a machine maintenance and smart diagnostics application.

For the case study we used an interface board equipped with sensors to monitor the temperature and vibration levels of different parts of a machine. The interface board makes use of digital, analog, serial and USB interfaces to capture data from a variety of sensors and systems. While modern industrial machines can directly communicate with the interface board (over digital, analog, serial or USB interfaces), many legacy machines make use of controllers that are impractical to access or digital communication is nonexistent. Therefore, the interface board makes use of sensors which are external to the legacy machines' control box.

We setup smart contracts for machine service and part replacements. These smart contracts act an agreement between the machine and the service/supplies vendors to either schedule a service request for the machine or place a part replacement order. Figure 5 shows the Python implementation of a controller service runs on the IoT device (Beaglebone SBC). The controller service constantly monitors the temperature and



```
from ethjsonrpc import EthJsonRpc
import serial
ser = serial.Serial('/dev/ttyACM0', 38400)

#Wait for sometime to initialize serial port
time.sleep(10)

# Connect to Blockchain network
c = EthJsonRpc('192.168.1.20', 8101)

tempCount = 0
vibrationCount = 0
tempThreshold = 200
vibrationThreshold = 100
tempCountThreshold = 1000
vibrationCountThreshold = 1000

while True:
  line = serialTTL.readline()
  data = line.split(',')
  vibration = data[0]
  temperature = data[1]

  if(vibration>vibrationThreshold):
    vibrationCount = vibrationCount+1

  if(temperature>tempThreshold):
    tempCount = tempCount+1

  if(vibrationCount>vibrationCountThreshold):
     #Send a transaction to requestService function of
     MachineService contract with Machine-ID
    c.call_with_transaction(c.eth_coinbase(),
    '0xc421d5e214ddb07a41d28cf89ee37495aa5edba7',
    'requestService(uint256)', [987])
    vibrationCount = 0

  if(tempCount>tempCountThreshold):
    #Send a transaction to orderPart function of PartReplacement
    contract with Part-ID & Qty
    c.call_with_transaction(c.eth_coinbase(),
    '0x20935bc944dba7b4d230b236edb36f8f31be3d9e',
    'orderPart(uint256,uint256)', [123,1],
    value=50000000000000000)
    tempCount = 0

  time.sleep(1)
```

Figure 5. Python implementation of a controller service running on the IoT device.

vibration levels of different parts of the machine (as sensed by the sensors on the interface board which is connected to SBC over serial port). Within the controller service, different rules are defined to determine if machine service request or part replacement order needs to be placed. For example, if the vibration levels of the machine go beyond a pre-defined threshold for a certain number of times, the controller service sends a machine service request. For sending the service request, a transaction is sent to the *request Service* function of the *Machine Service* smart contract between the machine and the service vendor. Similarly, if the temperature levels of a particular part of the machine go beyond a pre-defined threshold for a certain number of times, a part replacement order is placed. For placing an order, the machine sends a transaction to the *order Part* function of the *Part Replacement* smart contract between the machine and the part supplier by paying the part cost in crypto currency (Ethers).

For deploying the smart contracts (named *Machine Service* and *Part Replacement*), we used the Ethereum Go Client (*geth*) [14]. When a contract is deployed on the Ethereum Blockchain network, an address is assigned to the contract. For the case study we setup a private Ethereum Blockchain network. Any user on the Blockchain network who knows the contract address and the contract interface definition (ABI definition) can send a transaction to the contract. When the transaction is sent on the Blockchain network, it is combined with other pending transactions into a block. The miners on the Blockchain network validate the transaction and reach a consensus on the block that should be added next to the Blockchain (called the winning block). The new block is then broadcast to the entire network. The smart contract is executed when the transaction is mined.

8. Conclusion & Future Work

We presented a Blockchain Platform for Industrial Internet of Things (BPIIoT). The BPIIoT platform enables a marketplace of manufacturing services where the machines have their own Blockchain accounts and the users are able to provision and transact with the machines directly to avail manufacturing services.

The benefits of using Blockchain which make it suitable for Industrial Internet of Things are as follows:

- **Decentralized & Trustless**: Blockchain is a public ledger of all transactions on the network which is maintained by different decentralized nodes. Blockchain technology enables a decentralized and trustless peer-to-peer network where the peers do not have need a trusted intermediary for interacting with each other. Since a Blockchain network is not controlled by a central authority and all the transactions are verified and validated by a consensus among the peers, the peers do not need to trust each other.
- **Resilient**: Blockchain network is resilient to failures, as it is a decentralized peer-to-peer network with no single point of failure. The Blockchain itself is an immutable and durable ledger and the transactions once recorded on the Blockchain after a consensus among the peers cannot be altered or deleted.

- **Scalable**: Blockchain network is highly scalable in nature as it is maintained by a network of peers. The computing capability of the network scales up as more and more peers (or miners) join the network.
- **Secure & Auditable**: All the transactions in a Blockchain network are secured by strong cryptography. Furthermore, the transparent nature of the public ledger maintained by a Blockchain network makes it secure and auditable as everyone on the network knows about all the transactions and the transactions cannot be disputed.
- **Autonomous**: Blockchain can enable IoT devices to communicate with each other and do transactions autonomously as each device has its own Blockchain account and there is no need for a trusted third-party.

While the Blockchain technology looks promising for Industrial Internet of Things, there are various challenges that need to be addressed to ensure its widespread adoption. The key challenges are as follows:

- **CAP & Blockchain**: For distributed data systems, a trade-off exists between consistency and availability. These trade-offs are explained with the CAP Theorem, which states that under partitioning, a distributed data system can either be consistent or available but not both at the same time. Blockchain gives up on consistency to be available and partition tolerant. Blockchain is a distributed ledger which is eventually consistent, *i.e.* all nodes eventually see the same ledger. In Ethereum, the block-time (time after which a new block is mined) is roughly 17 seconds, which is much faster than Bitcoin which has a block-time of 10 minutes. A consequence of fast block-time is reduced security, therefore, many Blockchain applications require multiple confirmations for newly mined blocks to secure the transactions from double-spending. Faster block-time also leads to high number of stale blocks to be produced. Stale blocks are competing blocks produced by miners which do not contribute to the main chain. High stale rate reduces the security of the main chain [15]. To counter this problem, the "Greedy Heaviest Observed Subtree" (GHOST) protocol has been proposed [16].
- **Smart Contract Vulnerabilities**: Smart contracts can have software vulnerabilities which can be exploited by hackers. In June 2016, an attacker managed to drain more than 3.6 million Ether from the Slock.it backed Decentralized Autonomous Organization (DAO) into a "child DAO". The attacker managed this by exploiting a "recursive call bug" vulnerability in the DAO smart contract. Since smart contracts are meant to be agreements between transacting parties on a Blockchain and not legally enforceable outside the network, such attacks can put the organizations, miners and even the Blockchain network at risk [17].
- **Awareness**: Blockchain is a nascent technology and is mostly adopted in the financial sector (Bitcoin being the most popular application). Lack of awareness about the Blockchain technology in other sectors is affecting its widespread adoption.
- **Regulation**: Since Blockchain does away with the need for a centralized authority or a trusted intermediary for validating the transactions, there remain regulatory hur-

dles in the widespread adoption of the technology. New government and industry regulations are required for decentralized systems such as Blockchain. Furthermore, there is a need to ensure legal enforceability of smart contracts to avoid disputes among the transacting parties.

- **Privacy**: Since Blockchain is a public ledger and anyone can view all the transactions on a Blockchain network, there remain privacy concerns for the transacting parties.
- **Efficiency**: Since all the nodes in a Blockchain network perform the same computations in an attempt to mine the next block for the Blockchain, this is not an efficient approach. Due to this redundancy in execution, the contribution of an individual node to the overall network is very small even though the node may be performing very hard computations.

We described an implementation case study of the proposed BPIIoT platform based on Beaglebone Black single-board computer, interface-board based on Arduino Uno and the Ethereum Blockchain network. Future work will focus on implementation and demonstration of the BPIIoT platform for more realistic solutions such as on-demand manufacturing and device self-service.

References

[1] Bahga, A. and Madisetti, V. (2014) Internet of Things: A Hands-On Approach. VPT/Create Space Inc., Atlanta.

[2] Wu, D., Rosen, D.W., Wang, L. and Schaefer, D. (2015) Cloud-Based Design and Manufacturing: A New Paradigm in Digital Manufacturing and Design Innovation. *Computer-Aided Design*, **59**, 1-14.

[3] Xu, X. (2012) From Cloud Computing to Cloud Manufacturing. *Robotics and Computer-Integrated Manufacturing*, **28**, 75-86. http://dx.doi.org/10.1016/j.rcim.2011.07.002

[4] Colombo, A., Bangemann, Th., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., Jammes, F. and Lastra, J.L. (2014) Industrial Cloud-Based Cyber-Physical Systems. The IMC-AESOP Approach, Springer, Switzerland. http://dx.doi.org/10.1007/978-3-319-05624-1

[5] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

[6] Brody P. and Pureswaran, V. (2014) Device Democracy: Saving the Future of the Internet of Things, IBM. https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03620usen/GBE03620USEN.PDF

[7] Slock.it. https://slock.it

[8] TransactiveGrid. http://transactivegrid.net

[9] Filament (2016) Foundations for the Next Economic Revolution Distributed Exchange and the Internet of Things. https://filament.com/assets/downloads/Filament%20Foundations.pdf

[10] Christidis, K. and Devetsiokiotis, M. (2016) Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, **4**, 2292-2303. http://dx.doi.org/10.1109/ACCESS.2016.2566339

[11] Solidity Documentation. https://solidity.readthedocs.io

[12] Ethereum Homestead Documentation. http://www.ethdocs.org/en/latest/

[13] Wu, D., Thames, J.L., Rosen, D.W. and Schaefer, D. (2013) Enhancing the Product Realiza-

tion Process with Cloud-Based Design and Manufacturing Systems. *Journal of Computing and Information Science in Engineering*, **13**, 1-14. http://dx.doi.org/10.1115/1.4025257

[14] Ethereum Go Client. https://github.com/ethereum/go-ethereum

[15] A Next-Generation Smart Contract and Decentralized Application Platform (2016) https://github.com/ethereum/wiki/wiki/White-Paper

[16] Sompolinsky, Y. and Zohar, A. (2013) Accelerating Bitcoin's Transaction Processing Fast Money Grows on Trees, Not Chains. http://www.cs.huji.ac.il/~avivz/pubs/13/btc_scalability_full.pdf

[17] Siegel, D. (2016) Understanding the DAO Hack for Journalists. https://medium.com/@pullnews/understanding-the-dao-hack-for-journalists-2312dd43e993