*electronics*

MDPI

*Article*

# BAT—Block Analytics Tool Integrated with Blockchain Based IoT Platform

**Chathurangi Edussuriya [1], Kasun Vithanage [1], Namila Bandara [1], Janaka Alawatugoda [1], Manjula Sandirigama [1], Upul Jayasinghe [1], Nathan Shone [2] and Gyu Myoung Lee [2,*]**

[1] Department of Computer Engineering, University of Peradeniya, Peradeniya 20400, Sri Lanka; edussuriya.c@eng.pdn.ac.lk (C.E.); kas.vith@eng.pdn.ac.lk (K.V.); namilad@eng.pdn.ac.lk (N.B.); alawatugoda@eng.pdn.ac.lk (J.A.); manjula.sandirigama@eng.pdn.ac.lk (M.S.); upuljm@eng.pdn.ac.lk (U.J.)

[2] School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, UK; n.shone@ljmu.ac.uk

* Correspondence: g.m.lee@ljmu.ac.uk

Received: 10 August 2020; Accepted: 14 September 2020; Published: 18 September 2020

**Abstract:** The Internet of Things (IoT) is the novel paradigm of connectivity and the driving force behind state-of-the-art applications and services. However, the exponential growth of the number of IoT devices and services, their distributed nature, and scarcity of resources has increased the number of security and privacy concerns ranging from the risks of unauthorized data alterations to the potential discrimination enabled by data analytics over sensitive information. Thus, a blockchain based IoT-platform is introduced to address these issues. Built upon the tamper-proof architecture, the proposed access management mechanisms ensure the authenticity and integrity of data. Moreover, a novel approach called Block Analytics Tool (BAT), integrated with the platform is proposed to analyze and make predictions on data stored on the blockchain. BAT enables the data-analysis applications to be developed using the data stored in the platform in an optimized manner acting as an interface to off-chain processing. A pharmaceutical supply chain is used as the use case scenario to show the functionality of the proposed platform. Furthermore, a model to forecast the demand of the pharmaceutical drugs is investigated using a real-world data set to demonstrate the functionality of BAT. Finally, the performance of BAT integrated with the platform is evaluated.

**Keywords:** IoT; blockchain; data analytics; smart contracts; access management

## 1. Introduction

The Internet of Things (IoT) plays a significant role in the convenience of human daily life at present through various innovative applications and services. Further, it empowers the concept of autonomous systems creating a new social paradigm. The enormous amount of data generated by these services and systems are usually stored in on-premises servers and cloud servers depending on the context. However, these types of systems are vulnerable to several issues, including single point of failure, scalability problems, and Quality of Service (QoS) deficits due to its centralized architecture. Nevertheless, it is predicted that there will be 21 billion IoT devices at the end of 2020 [1]. Due to the above mentioned reasons, the traditional storage systems built upon the client-server architecture may not be able to withstand the growing large number of IoT devices and heterogeneous services. On the other hand, these systems are being used to communicate and store generic as well as application critical and privacy-sensitive data such as medical and financial records, Personally Identifiable Information (PII), and so forth [2]. Unauthorized access to these data could be result in aggressive advertisements to identity thefts to national security.

Users of IoT services provide personal information to service providers, intentionally or unintentionally and the use of this data is often decided by the service providers while users who have little, or no control over the access, or management of the data. For example, the data could be modified, or deleted from the systems without the concern of the users who have raised issues in many business models built upon IoT-platforms such as counterfeit of medicine in pharmaceutical supply chains. Owing to such data protection issues, GDPR (General Data Protection Regulations) [3] came into effect in the European Union (EU) region, trying to protect the data of the citizens with rights and regulations.

This paper presents an IoT-platform that uses blockchain with other state of the art technologies in solving the above identified problems. The decentralized and peer-to-peer architecture of the blockchain addresses problems related to centralized architectures and the tamper-proof mechanisms of the blockchain provide data integrity making the system resistant to unapproved modifications. Smart contracts are used to invoke the communication between IoT devices verifying the authenticity of data sources and a new access control mechanism is proposed to protect data and prevent unauthorized access. A smart contract [4] is a special code or program where there are few conditions mentioned. When the conditions are met by a specific user or a device, mutual authentication, or access can be granted.

The data produced by IoT-platforms can be used to derive useful information. However, there is no proper mechanism to verify the origin of the data, or the fact that the data have not been modified by unauthorized parties. We introduce a novel approach to create data-analysis applications using the data stored in the platform with the Block Analytics Tool (BAT). Hence, the data analysis applications can ensure the integrity, credibility, and authenticity of data that are used, for the analysis. Furthermore, the access control and authentication mechanisms of BAT ensures the security of the data.

The main role of the blockchain-based IoT-platform is the transaction processing. The performance of the transaction processing should not be effected with additional services provided by the platform. There must be a proper trade off between the analytical processing and transaction processing. Analytical processing is the processing and manipulation of stored data to obtain useful information. Hence, BAT is designed to process and acquire the blockchain data in an optimized manner without decreasing the performance of the transaction processing in the platform. Additionally, an index system called Block Index that is specifically designed for transactions is introduced to acquire data from the blockchain that reduces the cost associated with data retrieval. BAT facilitates decision making and predicting the future on trusted, and secured data produced by systems integrated with IoT devices without the involvement of third parties. We apply the proposed solutions on a real-life use case of a pharmaceutical supply chain to show the functionality of the proposed platform. Using BAT, a model to forecast the demand of pharmaceutical drugs is implemented using the data stored in the platform. Finally, we evaluate the performance of BAT integrated with our platform.

The main goals of the research can be summarized as follows. A blockchain-based IoT-platform is developed that provides countermeasures to scalability, security and privacy related concerns in IoT. A novel approach is proposed to perform data-analytics with the data stored in platform addressing the drawbacks such as the adverse effects on the transaction processing by the analytical processing, of the current state-of-the-art methods. The functionality of the proposed platform is presented through a use case scenario of a pharmaceutical supply chain and the functionality of BAT is shown through a development of a model to forecast the demand of pharmaceutical drugs using the data stored on the platform. A performance analysis is conducted to show the performance of the study through the use case scenario. The analysis of the results shows that countermeasures applied by the platform address the most of the concerns related to security, scalability and privacy. The experimental results show that BAT is able to maintain a proper trade off between transaction processing and analytical processing compared with the state-of-the-art approaches. Furthermore, the introduction of the Block Index increases the efficiency of acquiring data from the platform substantially.

The rest of the paper is organized as follows. Section 2 presents a literature review of the related work. The architecture overview of the proposed platform is explained in Section 3 and the design of BAT is explained in Section 4. The implementation of the use case scenario is presented in Section 5 and the results of the performance analysis of BAT integrated with the platform are elaborated in Section 6. Section 7 concludes the paper with the future directions of the study.

## 2. Related Work

Blockchain technology is a computing paradigm that is based on the distributed architecture for different parties to build trust in a trustless environment without third party involvement [5]. Bitcoin was introduced by Satoshi Nakamoto as a pure decentralized peer-to-peer electronic cash in 2008 that marked the initial implementation of the blockchain [5]. Blockchain is a distributed database or a ledger that contains timestamped records. These records are known as blocks protected by cryptography and linked to the previous block [6]. A transaction in the blockchain is verified by peers in the network. Without knowing the identity, a peer can verify a transaction and add it to the blockchain using the cryptographic hash of the block. History of transaction is visible via public keys but participants are anonymous. Peers need to verify a block before adding a transaction to the blockchain and distributed peers should agree on the order of the transactions before the block is added into the blockchain to maintain the integrity. This is known as the consensus mechanism. This ensures blocks are valid within the network. There are different types of consensus mechanisms used by the blockchain technologies such as Proof-of-Work (PoW) [7], Proof-of-Stake (PoS) [7], voting-based consensus [8].

Development of IoT-platforms using the blockchain technology has been attracted by many researchers and developers due to many reasons. The existing IoT-platform architectures are highly centralized and with the rapidly developing IoT services and applications, the centralized architectures will not be efficient or scalable to embrace the growing number of IoT devices [9]. The decentralized architecture of the blockchain will be efficient and it will be able to manage the increasing number of IoT devices and resist a single point of failure [10]. Jiang et al. [11] have focused on increasing the privacy of the users through multi-keyword search over the blockchain data on the blockchain. Access control is the selective restriction of access to the data [12]. There were data breaches when using apple Fitbit [13], where the data of the users were accessed by third parties without the consent of the users. With the blockchain technology the user can be sure that their data are not used without proper authorization [9]. The data become tamper proof that secures the integrity of data. FairAccess [14] is one of the platforms that improve access control by using a novel type of blockchain. Seyoung et al. [10] has also proposed a platform using the blockchain technology for the access management of IoT devices. The device acts as a node in the blockchain network in these studies [10,14]. However, when the device acts as a node, the performance of the blockchain reduces due to the power and performance constraints in the devices. Lei et al. introduced a blockchain platform [15] with a novel method of access control. The performance of the platform is comparatively high with the optimized execution procedure of the system. However, the data storage system and the querying mechanism of data are not optimized in this study.

Researchers have shown interests in utilizing the blockchain technology specifically for the industrial IoT (IIoT). BPIIoT [6] is a blockchain platform for IIoT that can be used to create distributed applications (DApps) for manufacturing. Using the DApps provided by the BPIIoT-platform, a machine can perform transactions with another machine, or a consumer without a third-party control. However, the focus is less towards the data management of the platform. There are numerous studies conducted to explore the data storage management of the blockchain technology. BeeKeeper [16] is an IoT-platform that is used for homomorphic computation [17] and secure storage. Homomorphic computation is the performance of computations on encrypted data [17]. The architecture of BeeKeeper is based on a beehive. The devices are considered as bees and the blockchain network together with the servers act as the beehive. The system creates more beehives with the addition of more devices to the network. Sapphire [18] is an IoT-platform that is proposed for data-storage management. The hashing mechanism used by the system is Location and Type Sensitive (LTS). Hence, this architecture is not suitable to be used in instances where geographical location is not particular. IoTA [19] uses a blockchain called Tangle and the performance of this platform is high. IoTA uses PoW as the consensus mechanism, that reduces the overall performance of the platform.

However, the studies conducted to build data-analytics applications using the data stored in the blockchain, still have many constraints and drawbacks. Studies have been conducted to improve the data analytics application of the blockchain [20]. The storage system of the blockchain is specifically designed to handle blocks of data and store the transactions as objects. The storage system is specially optimized to increase the efficiency of transactions in the blockchain. It is neither optimized to perform complex queries, nor as an efficient data retrieval schema. Hence, in most of the studies, off-chain database that contains the same data stored in the blockchain is used instead of using the on-chain database. As recommended by the Hyperledger Fabric developers [21], mirror storage facilities can be used to run concurrently with a blockchain storage system that replicates all the data in both locations for the data analysis. It is a wastage, as it doubles of all the resources used to store and analyze data. The mirror storage stores redundant data that will not be useful for any future reference. Moreover, updating the storage system at two locations concurrently reduces the processing time of a transaction that reduces the performance of the platform. In some studies, the on-chain network is used to store the security key and the data is stored in the off-chain database [22]. Hossein et al. [23] has taken the basic cloud architecture and decoupled the data plane and control plane of the architecture and restructured the architecture to be used as an IoT-platform. Data plane has the data storage system whereas the control plane, manages the data stored in the data plane [23]. The usage of the blockchain technology in the control plane has enabled the control of data and access control by the proposed system itself without a centralized authority. The data is stored in a separate storage and the hash pointer is stored at the blockchain. In this mechanism, every time a transaction is processed, the data stored in the off-chain network has to be retrieved to identify the state of the block. For instance, in supply chain transactions, the current owner of the specific object has to be identified to perform the transaction between the current owner and the new owner of the object. For this purpose, the off-chain database has to be queried. Hence, the performance of transaction processing will be reduced drastically. In our study, we introduce a new approach to mitigate these constraints and drawbacks called BAT.

A pharmaceutical supply chain is used as the use case scenario in the study. Drug counterfeit is a severe problem in pharmaceutical supply chains. This occurs due to the lack of transparency in the transportation of drugs through the supply chain. Certain medicines require to have proper conditions maintained. These details remain hidden and there is no trusted method to monitor these conditions. According to the World Health Organization, more than 10% of medicines worldwide are counterfeited [24]. Manipulating the expire date, producing drugs with no active chemical ingredients [25] are some instances where drug counterfeiting occurs. After distributing these ill-treated drugs, users are unable to identify these counterfeit medicines. There is no proper mechanism to verify the integrity of whether the original package is distributed by the third party logistics company [24]. Most of the above-mentioned problems can be addressed by establishing trust between the parties in the supply chain. A blockchain based IoT-platform can be used to process transactions that occur in the supply chain and the implementation is presented in the Section 5. Figure 1 shows the scenario of the use case chosen. In the use case, the production companies produce medicine, Third Party Logistics (3PL) supply the medicine to the warehouse. Through the warehouse, the medicine is distributed to the pharmacies or the issuers. Most of the time drug counterfeiting occurs through the suppliers and the warehouse.

Blockchain acts as a data storage system for transactions as all the transactions occured in the supply chain are recorded in the blockchain. These data can be used for data analytics of the supply chain. Maintaining the supply and demand in pharmaceutical supply chains is very important. Demand forecasting in the pharmaceutical industry is critical as the availability of drugs at the needed time impacts on patient's life [26]. Furthermore, the demand for drugs by different pharmaceutical companies is a complex combination of the necessity of drugs, shelf life, regulations and the cost associated with drugs. The consumption method [27] of the forecasting of drugs is the usage of historical data of past consumption of drugs. As the data is recorded in the blockchain in a well-ordered manner, it can be used to predict future requirements of drugs. Functionality of the BAT was experimented with the use case scenario of developing a demand forecasting model for pharmaceutical drugs as shown in Section 5.2.

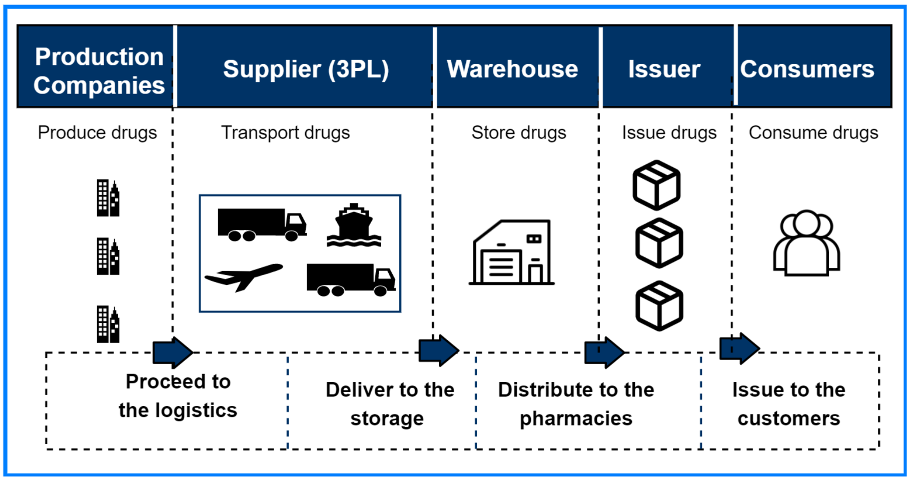


**Figure 1.** Use case scenario.

## 3. Architecture Overview of the Proposed Platform

Figure 2 shows the architecture overview of the platform. A modular architecture is adopted with a layered structure that ensures that each layer could be designed separately without altering the other layers. The architecture is divided into 4 layers. Each layer is interfaced with the other layer through a communication medium.

The device layer that consists of the sensors and actuators, is connected through a local gateway to the network layer. The network management of the sensors and actuators in the device layer is performed by the network layer. Basic security protocols are implemented for the transmission of data and control between the device layer and blockchain such as data encryption to prevent unauthorized listening and understanding of the content. In the proposed system, MQ Telemetry Transport or Message Queuing Telemetry Transport (MQTT) [28] can be used as the protocol for communication of IoT devices as it is lightweight and more suitable for the communication between constrained devices as it has a small header [29]. Furthermore, the network layer uses Transport Layer Security (TLS) [30] for the encryption of data. The network layer is connected with the blockchain network through a message broker. In our implementation, MQTT broker is used as the message broker.

Blockchain is the main actor in the platform. The service layer is interconnected with the blockchain. Wallet services help the platform in the process of identity management. Wallets are produced by each certification authority of the blockchain network. A wallet contains digital certificates and security keys that can be used for the identification of a component connected to the network. The particular organization that issued the certificates to a user validates the certificates upon request. This ensures that the identity management service of the platform is decentralized between the organizations of the blockchain. Furthermore, each component connected to the platform can be granted with levels of privileges and access control. A user can have the privileges of an admin, writer, or a reader [31]. Hence, a user will have only the essential set of permissions, that is, a client user (reader) of the platform will not be able to alter the configurations of the platform.

The transaction processing is one of the major functions of the platform. A consensus mechanism is used for the ordering and validation of transactions. The platform uses a permissioned voting-based consensus mechanism as explained in Section 2. An endorsement process [32] is performed for the validation of the blocks. The endorsement policy defines that organization needs to approve the transaction. In our proposed platform, all the organizations connected to the network should validate the transaction. Event management resides inside the service layer. In the proposed platform, the data are requested from the device layer with a triggering of an event.

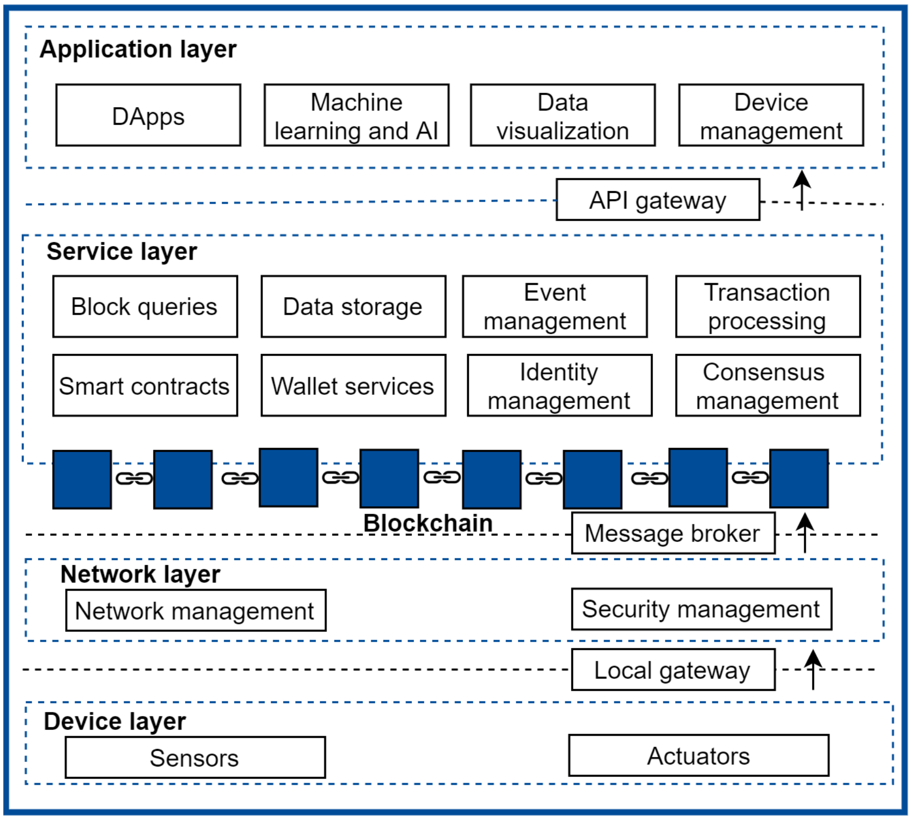


**Figure 2.** The architecture overview of the proposed blockchain based Internet of Things (IoT)-platform.

Blockchain stores data of all the transactions processed in the platform. Data related to peer to peer communication are stored as blocks without the control of a central authority in the data storage. Hence, the blockchain acts as a data warehouse [33] that stores data from different sources (e.g., IoT devices, DApps data, management data, etc.). These data are very useful especially for industries and business. With the proposed system, data can be retrieved and visualized easily specially for business analytics. A novel approach to query the blockchain is introduced in Section 4. On top of the services provided by the blockchain, the application layer is visible that exposes these services to the external users. This layer is interfaced with the service layer using an API gateway. The transaction processing and interacting with front-end applications are developed through DApps.

In the study, a special tool called BAT facilitates the development of Machine Learning (ML) and Artificial Intelligence (AI) applications using the data stored in the blockchain, as explained in Section 4. Each IoT-device is registered before data transmission in the platform. Device management in the application layer is used to handle IoT devices. The registration and management is performed through a smart contract specifically created for that purpose. Each device is also provided with a wallet to prove its identity. Hence, this mechanism ensures that unauthorized devices cannot communicate in the system. This provides a solution to the problem of integrity of IoT-devices.

Figure 3 shows how the different services of the layers in the architecture collaborate together to perform a transaction. First of all, when a user wants to interact with the proposed platform, the user should first provide the identification certificates that are stored in the wallet of the user to the blockchain network. The wallet is issued to a user by a specific organization established in the platform. This process is done through the Application Programmable Interface (API) gateway connected to the REpresentational State Transfer (REST) API [21]. If the user fails to produce the wallet containing proper certificates, the users will not have access to the platform.

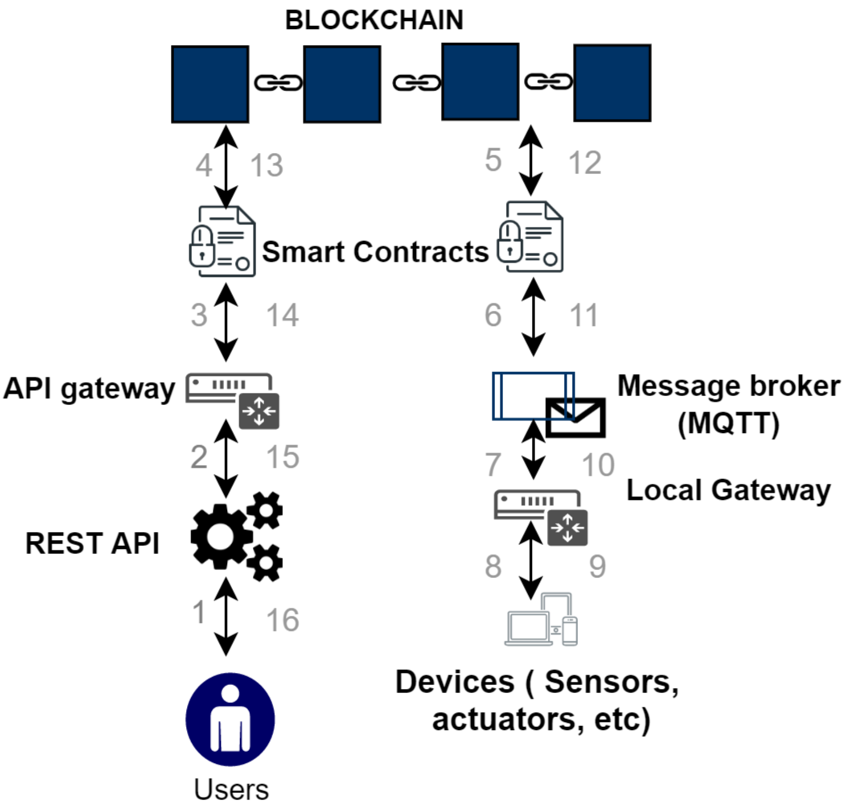


**Figure 3.** Transaction execution procedure.

After the authentication, the request for the specific transaction, or the process is submitted to the API gateway through the application. The smart contract will be invoked once the request is received by the blockchain network. From one smart contract to the other the specific logic function is changed. The use of a smart contract removes the necessity of a centralized control, or a third-party access of the system. If the necessary conditions that are specified by a smart contract, are fulfilled, the endorsement peer will approve the transaction and commit the transaction. The other entities of the network are informed about the transaction through the Orderer [21] of the network and all the peers will update their ledgers about the transaction. Orderer is the entity that is responsible to maintain consistency between the blockchain organizations (e.g., peers, ledgers) [21]. Then, the response of the transaction is sent back to the application through the gateway. If the particular smart contract requires data readings of the IoT-devices connected to the network, an event is triggered through the chaincode and the network subscribes to the message broker (MQTT broker) of the system. Through the local gateway, the sensors and actuators will publish the encrypted data.

## 4. Block Analytics Tool (BAT)

In this section, a novel approach called BAT is proposed. The tool primarily enables analytical processing of the data stored in the blockchain and also addresses the drawbacks in state-of-the art approaches, discussed in Section 2. An overview of the BAT is shown in the Figure 4.

We aim to ensure the privacy and security of the data that is being used for that analysis. Hence, all communications that take place through BAT also adopts the secure features used in the blockchain based IoT-platform. Smart contracts are used to establish the communication between the platform and BAT. Wallet services and identity-management policies are used to ensure that the unauthorized parties cannot use BAT for analysis. BAT acts as an Extraction Transfer Load (ETL) [34] tool for the blockchain. A system administrator provides a configuration file to BAT that specifies the data required as well as about the components of the BAT such as the type of Blockchain Data Cache (BDC) required.

Only the data requested by the user are retrieved, reducing the cost associated with storing redundant data. The overall execution procedure of BAT is shown in the Figure 5.

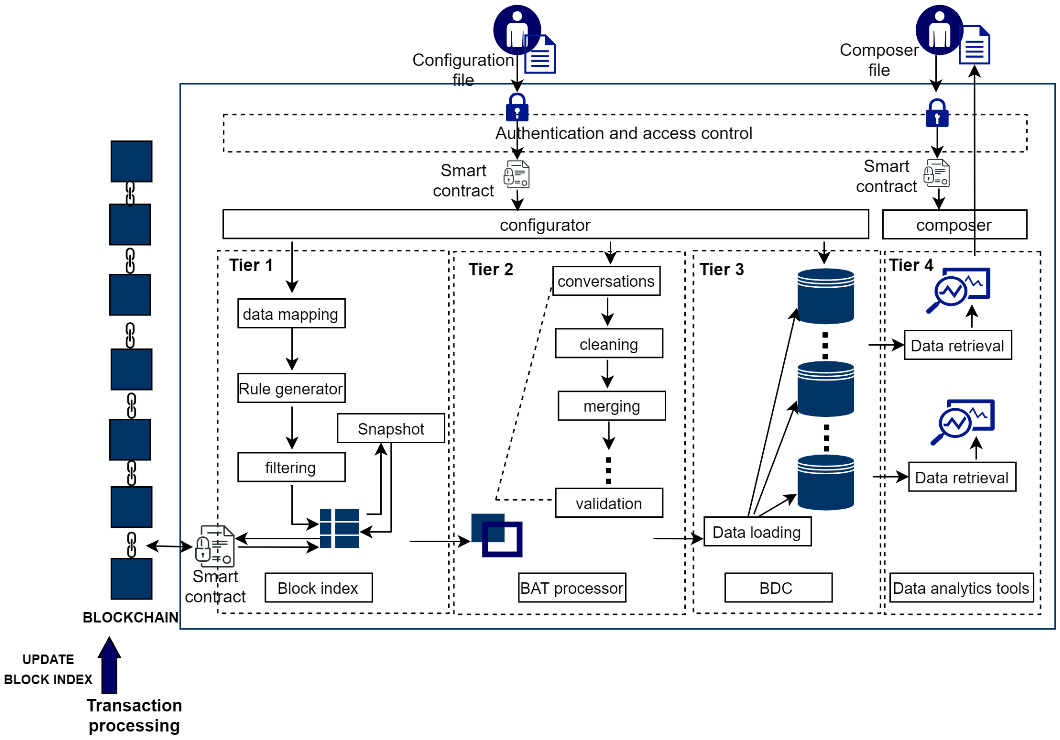


**Figure 4.** Overview of the Block Analytics Tool (BAT) integrated with blockchain.

There are four main tiers in BAT that have separate functionalities and these are explained in detail along with the execution procedure. Tier 1 is the extraction of data from the blockchain. Extracting data from the blockchain acts as a bottleneck due to the effect of data extraction on the transaction processing. As mentioned earlier rich queries can degrade the performance of the blockchain system. Furthermore, to query a single block, all the blocks in the blockchain have to be searched. Hence we use a novel approach to extract data from the blockchain through an index specifically created for transactions occurring on the blockchain. A special index called Block Index is proposed to increase the efficiency of the data extraction process. Block Index acts as a filtering process for the data.

The blockchain technology is used to handle transactions between two parties. A unique ID is given for all the transactions recorded in the blockchain. In a transaction the ownership of this specific object changes while the other details about the object remain the same. For example, in vehicle trading between two parties, the ownership of the vehicle changes while specifications about the vehicle, such as dimensions, engine capacity, fuel capacity remain the same. Moreover, if we want to get the details about a particular object, the search pointer goes through many blocks that contain the same details that increase the search time to a large extent. To address these drawbacks, we propose the Block Index system.

Let’s define objects, parties, and transactions as below;

$$\begin{aligned} Objects(B_x) &= B_1, B_2, \ldots\ldots, B_m \\ Parties(p_m, p_n) &= P_{1,2}, P_{2,3}, \ldots, P_{n-1,n} \\ Transactions(t) &= B_1 P_{1,2}, B_2 P_{2,3} \ldots, B_n P_{n-1,n}. \end{aligned}$$

In here, the transactions occur related to object 1 between party 1 and party 2 is represented as $B_1 P_{1,2}$. When object 1’s transaction occur between party 2 and party 3 the details about the object will remain the same where as the current owner changes from party 2 to party 3. This relationship is considered and the Block Index is created as a matrix representation of the transactions. Each row in the Block Index represents the transactions related to one particular object. The columns represent the transactions between different parties. The chain of transactions related to the objects between different parties is mapped into a matrix representation.

$$\begin{bmatrix} & P_{1,2} & P_{2,3}. & . & . & . & . & P_{n-1,n} \\ B_1 & B_1P_{1,2} & B_1P_{2,3}. & . & . & . & . & B_1P_{n-1,n} \\ B_2 & B_2P_{1,2} & B_2P_{2,3}. & . & . & . & . & B_2P_{n-1,n} \\ . & & & & & & & \\ . & & & & & & & \\ . & & & & & & & \\ B_m & B_mP_{1,2} & B_mP_{2,3}. & . & . & . & . & B_mP_{n-1,n}. \end{bmatrix}$$

The blockchain records the data of a transaction as soon as it occurs. The details about the transaction is mapped into the Block Index easily. The implementation of the addition of one block in to the Block Index is shown in the Algorithm 1. Through this representation, the data extraction processed can be performed more efficiently. If we assume that the blockchain recorded the data about transactions as below:

$$B_1P_{1,2} \Rightarrow B_2P_{1,2} \Rightarrow B_2P_{2,3} \Rightarrow B_1P_{2,3} \Rightarrow B_3P_{1,2} \Rightarrow B_4P_{1,2} \Rightarrow B_3P_{2,3} \Rightarrow B_3P_{3,4} \Rightarrow B_4P_{2,3} \Rightarrow B_1P_{3,4}$$

$$\begin{bmatrix} & P_{1,2} & P_{2,3} & P_{3,4} \\ B_1 & B_1P_{1,2} & B_1P_{2,3} & B_1P_{3,4} \\ B_2 & B_2P_{1,2} & B_2P_{2,3} & \\ B_3 & B_3P_{1,2} & B_3P_{2,3} & B_3P_{3,4} \\ B_4 & B_4P_{1,2} & B_4P_{2,3} & \\ . & & & \end{bmatrix}$$

If the details about the object related to the 10th transaction $t_{10}$. Then, instead of searching through 10 transactions, the Block Index gets the required details by only searching the first column ($P_{1,2}$) as all the other column have the replication of the same details. In the worst case scenario, this reduces the data extraction time from $O(mn)$ to $O(m)$ where $m$ is the number of objects and $n$ is the number of parties.

Through the configuration file, a user can provide the necessary data to be extracted. Then the specified data are mapped to be used by the Block Index architecture. For instance, if the user wants the data about the transactions between party $P_m$ and $P_n$ the data from the $P_{m,n}$ has to be retrieved. After mapping the data, rules are created to retrieve data from the blockchain. These rules are the queries that are used to get data from the blockchain storage system. Through the Block Index the data are extracted. The communication between the platform and BAT takes place through a smart contract.

The extraction process takes place through pagination to ensure the consistency of data and to avoid the overflow of memory. After the initial extraction, BAT keeps track of the last set of data that has been queried from the platform and this is kept as a snapshot of the data retrieved. For instance, if the details about the block $k$ ($B_k$) were queried in the last extraction, BAT keeps a tag stored to assure that in the next data extraction time, the same data is not copied that reduces the data extraction time. As soon as a transaction occurs in the platform, an event is triggered to update the Block Index. Updating the Block Index takes a small amount of time, whereas in the current approaches of analytical processing, the latency associated with the transaction processing is very high. The performances are compared in the Section 6.

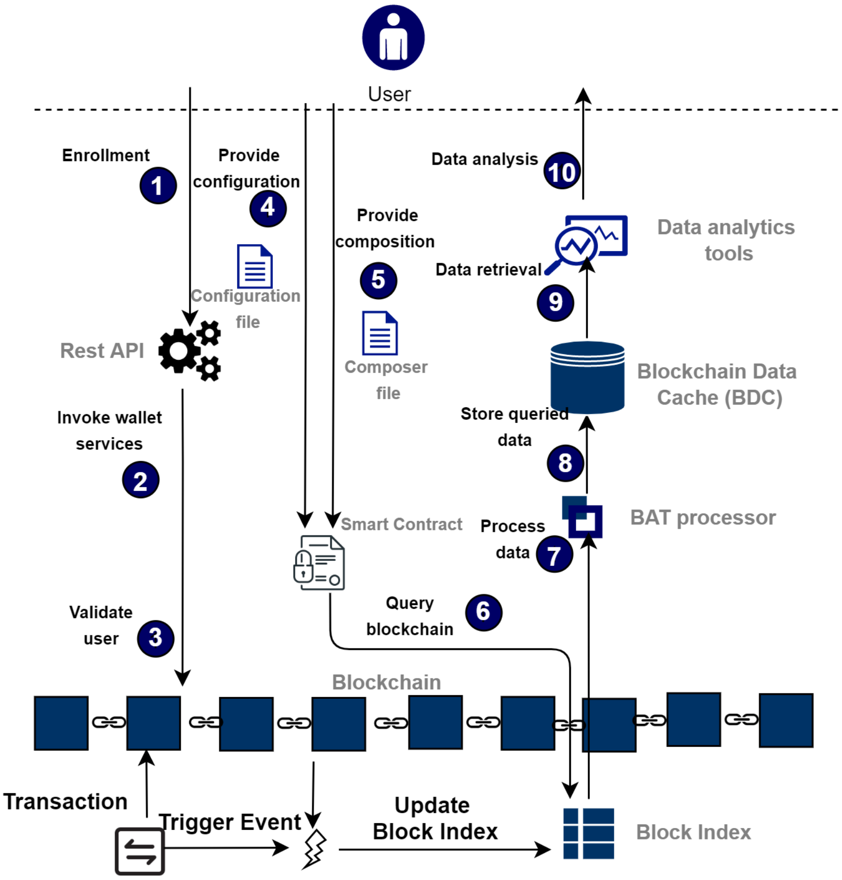


**Figure 5.** Overall execution procedure of BAT integrated with the blockchain.

**Algorithm 1** Pseudo code to add a block to the Block Index.

```
Input: x[ID] - ID of new block
Output: column of the index, row of the index
Initialization:
nRows0 = number of rows in column 0
nColumnsi = number of columns in row i
IndexColumn = 0
for $i = 0$ to $nRows0$ do
    if ($i[ID] = x[ID]$) then
        for $IndexRow = 0$ to $nColumnsi$ do
            IndexColumn = IndexColumn+1;
        end for
    end if
end for
return $IndexColumn, IndexRow$
```

Algorithm 1 shows the pseudo code of the Block Index. When a transaction occurs, the Block Index is updated through the chaincode. If the block contains the details about a newly created object, the Block Index stores the transaction of the object under the column 0. When the object undergoes another transaction, the transaction is recorded in the second column through the chaincode. The querying process uses this Block Index created to retrieve the information.

Tier 2 is the data-transformation phase. After the data-retrieval phase, the queried data are transformed to be stored in the BDC. The process of transformation of data to the user required data-type that is termed as the Object Relational Mapping (ORM) [35], is conducted by the BAT Processor. Furthermore, BAT processor performs transformation functions on the data for the data standardization as explained below. These transformations are taken place in the processor as separate operations. One object is considered as an entity, or a column where the fields are considered as rows in the transformation. These operations can mainly be categorized into two as column wise operations and row wise operations. The semantic representation of the data set can be shown as follows. Data set = $D$, Entity object (row) = $e$, Field (column) = $f$, attribute value = $val_{e,f}$.

Following shows few of the operations that are carried out by the BAT processor. Column wise operations such as Conversations (Unit conversations...): $\forall val_{e,f} \in f, c = constant \Rightarrow val_{e,f} \times c$, Merging fields (join): $\forall val_{e,f_x} \in f_x, \forall val_{e,f_y} \in f_y \Rightarrow val_{e,f_{x,y}} = val_{e,f_x} \bowtie val_{e,f_y}$ and Row wise operations such as cleaning (removing null rows): $\forall e = \emptyset, e \in D \Rightarrow D = D - e$, cleaning (removing duplicate rows): $r_x, r_y \in D, r_x = r_y \Rightarrow D = D - r_x$.

The user can explicitly mention the operations and conditions that the data set must go through to transform the data. BAT creates a precedence process of operations must be followed to ensure that the data is processed in linear time. The column wise operations are taken place first followed by the row wise operations. Functions such as merging functions are at the end of the process. The algorithm of how BAT processor transform the data is shown in Algorithm 2.

Tier 3 is the data storage phase. After processing the data, it is loaded in to the BDC and it acts as a data warehouse for the blockchain data. The user has the capability to determine the type of database to be used by BAT between relational and non-relational databases. After the initial loading of data to the BDC, the data can be extracted in incremental phases to BAT. BAT make sure about the maintainance of the Atomicity, Consistency, Isolation, and Durability (ACID) properties [36] between the BDC and the blockchain.

**Algorithm 2** Algorithm for operational precedence process of BAT processor.

```
Input: D (Data set), Operations
Output: ProcessedD - processed data set
Initialization:
OpcolF = Operations and columns where operations needed to be performed
Oprows = Operations on rows
ProcessedD = D
for $opcol = 0$ to $OpcolF$ do
    ProcessedD = opcol(ProcessedD)
    for $oprow = 0$ to $Oprows$ do
        ProcessedD = oprow(ProcessedD)
    end for
end for
return ProcessedD
```

Tier 4 is the data analysis phase. The data analytics tools and frameworks (Tensorflow [29], Pytorch [30], etc.) as specified by the user through the configuration files, can be integrated with BDC. Unlike the blockchain data storage system, the tools can perform rich queries, preprocess data stored in BDC. For ML developments, latency and throughput plays a critical role. Usage of BDC instead of acquiring data directly from the blockchain reduces the latency and increases throughput. Different users can make use of the same BAT instance. The users can provide a composer file that specifies the type of data required by different users and they will be available at the data analysis tools specified by the user.

All communication between the tool and the users take place through the access control and authentication process of BAT. The users should have a valid wallet issued by the blockchain to use BAT. Furthermore, the access control policies ensure that only the users who have the read privileges can use BAT tool. Smart contracts are used when interacting with different users. These smart contracts validate the configuration files and the composer files provided by the users and develop the instance of BAT or data acquisition process as needed. The communication with the blockchain takes place through a smart contract. This smart contract ensures the authenticity and integrity of data that is being used for the analysis. The smart contract checks whether the data are generated or stored in the platform and the it also makes sure that the data have not been modified by any unauthorized parties.

## 5. Case Study

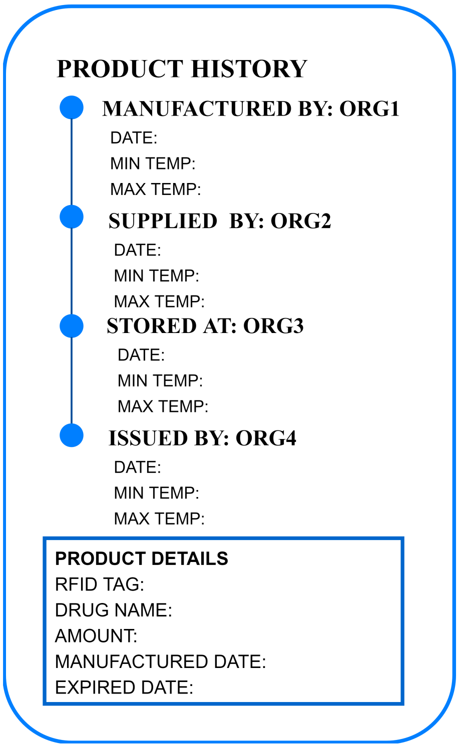
This section explains about the implementation of the blockchain-based IoT-platform as well as the integration of BAT to the platform through the use case scenario of the pharmaceutical supply chain as explained in Section 2. For this study, Hyperledger [21] is chosen as the candidate blockchain technology as it is the most suitable type of blockchain for handling business logic currently [31]. RFID tags are used to prevent tampering of the expire date, manipulation of the dosage information in the medicine package. Drug manufacturers add initial package data into the blockchain and RFID tags that can be used to verify the details at any given point using the proposed platform. Sensors can be used to ensure that the drugs were transported in the correct condition. For example, temperature sensors are used to check whether the temperature of drugs was maintained.

### *5.1. System Architecture of the Case Study*

The implementation and analysis of the proposed system was performed in the gcloud virtual machine instance (configuration of Name- c2-standard-4, Zone- us-central1-f, vCPUs- 8, Memory-30GB) [37]. RFID tags resides in the device layer of the blockchain-based IoT-platform. Through a gateway it is connected to the network layer. MQTT broker is used to transfer data from the RFID tags as explained in Section 3. The architecture of the blockchain used in the use case scenario are explained as follows. The main actors of the scenario are the supplier, warehouse and the issuer organizations. The network is initiated by the warehouse organization and the initial configuration of the network is configured by the warehouse entity. The network is controlled according to the rules imposed by the network configuration. The channel order is maintained by the orderer connected to the channel. The channel is governed by the channel configuration that has the policies related to all 3 entities. Separate certificate authorities are maintained by each entity for the validation of the transactions processed in the blockchain. Each organization consists of 2 peers and 1 peer is used as the anchor peer that is used to communicate with the other entities. In each peer, a copy of ledger containing details about all the transactions is installed.

The service layer consists of services such as smart contracts, data storage management, transaction processing, identity management. The peers contain copies of smart contracts installed in the platform. There are 2 smart contracts called create-batch and transfer-batch, that are utilized by the applications in the platform. A manufacturing company uses create-batch smart contract to create a batch to be delivered to one issuer company. The details about batch of drugs such as name, dosage, quantity, manufactured and expiry date are added as a block to the blockchain. An RFID tag is assigned for this particular batch. Transfer-batch smart contract is used when a batch is transferred from one entity to the other.

Application layer consists of DApps and Data visualization and Device management. Four different applications are created as DApps that are used by different organizations for transactions and processing of data. They are used by production companies, suppliers, warehouse and issuers for the transactions. Through another application, an end user can examine where a specific batch is located on that occasion. Eventually, the end user at the issuer can see how the batch has been transported, stored and issued by the organizations while maintaining the favorable conditions for batches with complete transparency. Figure 6 shows the user interface of the application at the end of transferring a batch through the supply chain.



PRODUCT HISTORY

MANUFACTURED BY: ORG1
DATE:
MIN TEMP:
MAX TEMP:

SUPPLIED BY: ORG2
DATE:
MIN TEMP:
MAX TEMP:

STORED AT: ORG3
DATE:
MIN TEMP:
MAX TEMP:

ISSUED BY: ORG4
DATE:
MIN TEMP:
MAX TEMP:

PRODUCT DETAILS
RFID TAG:
DRUG NAME:
AMOUNT:
MANUFACTURED DATE:
EXPIRED DATE:

**Figure 6.** User output of product history at the end of transferring a batch through the supply chain.

### *5.2. Demand Forecasting of Pharmaceutical Drugs*

With the help of hospital prescribing dispensed in the community data set [38], batches of drugs were created. The data about hospital prescribing data related to the Manchester University NHS Foundation [38] were used for the study. Batches were added in a manner where batches were transferred through the supply chain at different stages.

The transactions between different organizations can be shown as follows. production to supplier = $P_{pr,s}$, supplier to warehouse = $P_{s,w}$, warehouse to issuer = $P_{w,i}$, issuer to patients = $P_{i,p}$, Different batches = $B_1, B_2, \ldots$ In the representation, number of columns = number of transactions between different organizations, row = history of a batch, $P_{i,p}$ column = batches utilized by the end users.

$$\left[\begin{array}{ccccc} & P_{pr,s} & P_{s,w} & P_{w,i} & P_{i,p} \\ B_1 & B_1P_{pr,s} & B_1P_{s,w} & B_1P_{w,i} & B_1P_{i,p} \\ B_2 & B_2P_{pr,s} & B_2P_{s,w} & & \\ B_3 & B_3P_{pr,s} & B_3P_{s,w} & B_3P_{w,i} & B_3P_{i,p} \\ B_4 & B_4P_{pr,s} & B_4P_{s,w} & B_4P_{w,i} & \\ B_5 & B_5P_{pr,s} & & & \end{array}\right]$$

A block contains details about RFID tags, drug name, dosage, quantity, cost, organization (e.g., production, supplier, etc.), temperature sensor readings, transaction time, manufactured and expired dates. During a transaction the organization, temperature sensor readings change while the other details remain the same. The Block Index was updated with the addition of batches to the platform. The above matrix representation shows part of the Block Index that shows the state of transfer of batches through the supply chain. According to the Block Index, batch $B_2$ is now stored at the warehouse. The transaction ID is used to map the Block Index to the blockchain.

For the demand forecasting analysis, the details about issued drugs such as drug name, quantity, cost and issued date of batches are required for the analysis. According to the Block Index designed, the details about issued batches to the patients are available in the 4th column. Using the Block Index, the specific batches could be extracted without querying the full blockchain. If the Block Index was not used for this purpose, the querying is more complicated with more functions and algorithms. In the implementation, CouchDB [39] database is used as the state database. CouchDB has a default indexing system that can be used to query the database easily. The proposed Block Index makes use of this system.

After retrieving data from the blockchain, the data are processed by the BAT processor. The data are cleaned, and validated according to the operational-precedence process. In the Hyperledger, data are stored as json objects. If the user requests the data to be stored in a relational database such as MySQL, the ORM process has to be performed where data is transformed from json objects in to tabular schema. The BDC stores the processed data. BAT supports both on-premise and cloud-storage systems. In the use case scenario, non-relational database MongoDB [40] was used as the candidate on-premise database for BDC. For the data analysis application, the cumulative quantity and costs of drugs in a monthly basis are required. If the data from the blockchain are directly extracted, the performance of the platform is reduced drastically with the complex query as the transaction processing is effected. However, with the BDC, the query can be performed easily without effecting the performance of the platform.

In the case study, Tensorflow was used as the ML tool. The demand analysis of supply chain data is a time series analysis as the seasonal weather patterns, average price is affected by the time. Pharmaceutical drugs show correlations between each other as many drugs are prescribed as a set for a specific disease. While preserving these correlations, the time series model was built using ML algorithms, xgboost regressor [41] with SVR (Support Vector Regression) [42] as the base estimator.

Figure 7 shows the demand prediction and actual values of total quantity of certain types of drugs for the month October of 2019 for the hospitals in Manchester University NHS Foundation [38]. The model was built with the an average accuracy of Root Mean Square Error (RMSE) value of 0.5213. The model utilizes previous 12 months of data while analyzing the correlations of the drugs and the demand of drugs for the next month can be predicted. For an instance, the drug quantity needed by "Amlodipine-Tab 5mg" of drug code "0206020A0" can be predicted with an RMSE of 0.3134.

The status of the model built is saved and when the model is required again, the composer file can be given to BAT in the same manner and the analysis could be conducted with the newly added blocks. Hence, the new data added will also be included in the learning process of the model, that increases the performance of the model built.

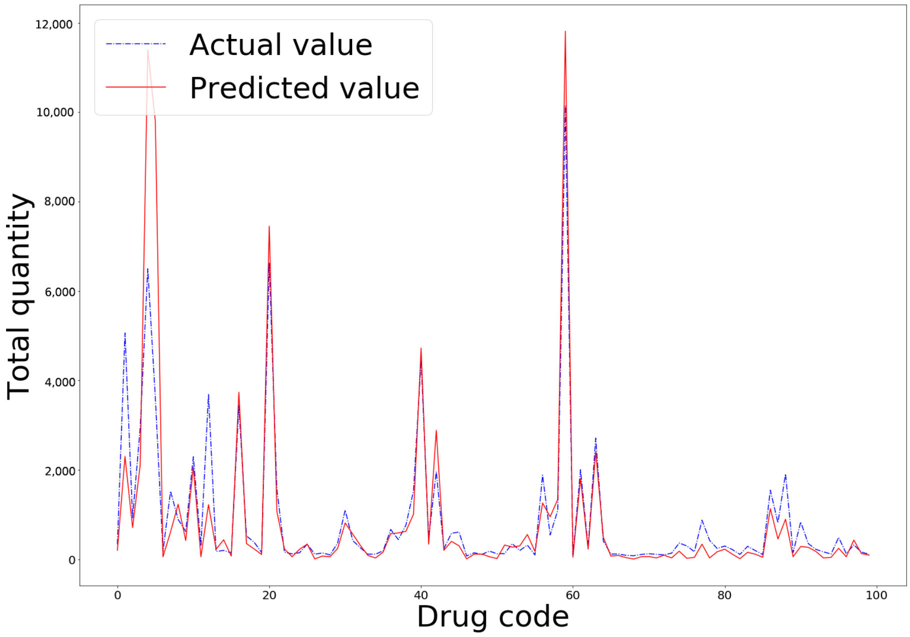


**Figure 7.** Predicted and actual value comparison for the demand forecasting.

## 6. Performance Analysis

In this section, we present the analysis of the performance of BAT integrated with the platform. Figure 8 shows the total transaction processing time taken by different applications for creating and transferring batches with IoT-devices connected. The transaction processing time increases when the number of blocks in the blockchain increases and highest transaction time is taken for the creation of the new batch that is the transaction between production and supplier companies. This is due to the additional device registration time related to RFID and temperature sensors.

The following are few mechanisms that ensure the authenticity and integrity of batches while assuring the quality.

- Adding counterfeit batches, or changing the details (e.g., expiration date, dosage, etc.): Counterfeit batches cannot be added as the RFID tag is cross checked with the first block of batch (production to supplier transaction). Furthermore, the smart contract rejects the transaction if the details produced does not match with the details of the first block of batch.
- Replacing contents of batch: The RFID tags are attached to the boxes in a way that the system is alarmed, if batch is broken/damaged. Hence batch is accepted.
- Impersonate the authorities while processing transactions: Every person performing a specific transaction owns a wallet issued by the particular organization. If the person does not own a legitimate wallet, the transaction is rejected.
- Unauthorized access of data: In device management, usage of BAT and other applications are performed through smart contracts, wallet services and the access management policies of the platform make sure that only authorized parties are allowed to access data.
- Modifing, or adding data while using BAT: BAT only accesses one ledger in the blockchain and other peers and ledgers are not notified about the transaction. Hence, the endorsement fails rejecting the modification.

The implementation of the demand forecasting of pharmaceutical drugs in Section 5.2 shows the basic functionality of BAT integrated with the platform. BAT uses authentication and access control mechanisms when interacting with users and while communication between the blockchain. Owing to these reasons, developers can ensure the privacy and security of the data as well as the authenticity of the data used in the analysis. The conventional method of the usage of the off-chain database and the introduced novel-approach is compared to present the performance between the two methods. The main usage of the blockchain based IoT-platform is the transaction processing. Hence, the performance of the transaction processing should not decrease with the addition of other services in to the platform. When BAT is used for analytics, the overall performance of the platform related to transaction should not be effected.

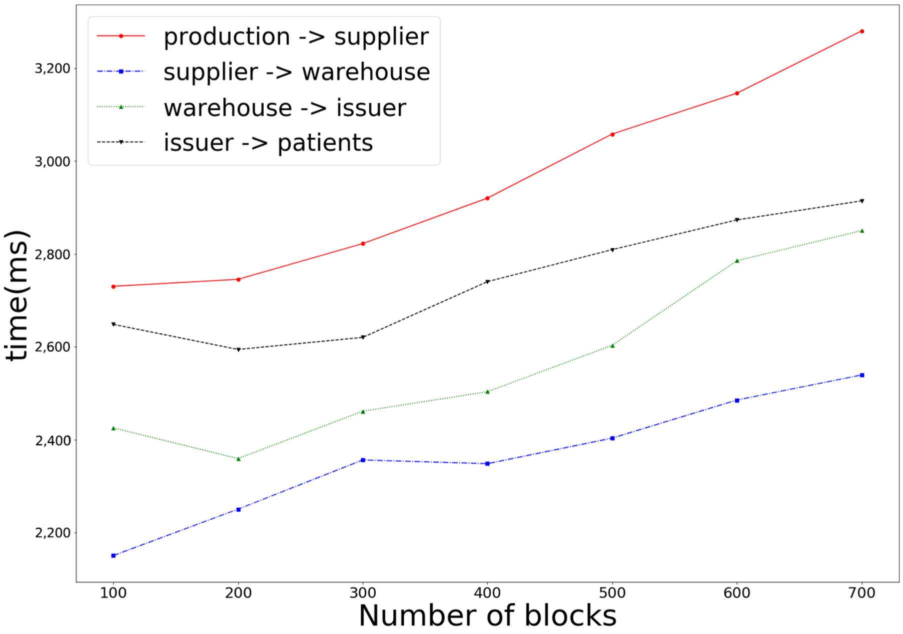


**Figure 8.** Total transaction processing time.

Figure 9 shows the comparison of the effect of event processing on transaction processing when using BAT and the conventional method of usage of an off-chain DB.

When transactions take place, the total processing time of the transaction is the addition of the transaction-processing time and the event processing time. In BAT, only the Block Index is needed to be updated whereas the data is duplicated into an external database in the conventional method. This increases the event processing time exponentially. On the other hand, a constant time between 0–5 ms is used to update the Block Index. Hence, the effect of the analytical processing on the transaction processing remains constant at almost 0 ms when using BAT whereas in the conventional method, the effect on the transaction processing increased exponentially with the increase of the size.

Transaction throughput is the number of transactions that can be processed in a second. Even though the total throughput decreases with the increase of the block size, the throughput of the conventional method is comparatively low compared with BAT as shown in Figure 10. Furthermore, the off-chain database duplicates all the data in the blockchain creating redundant data whereas the size of the Block Index will remain less than 1 MB most of the time. The storage cost associated with the current approach is drastically high as it stores lot of redundant data.

The comparison of the data acquiring process using BAT and the conventional process is compared in Figure 11. BAT utilize the Block Index in the process of acquiring data whereas in the conventional process, the normal querying architecture is used. In the figure, batches issued to the users mean retrieving data of the 4th column of the Block Index. Selecting quantity for the batch x is getting data about a batch as the ownership is the only variable that changes through a row. Hence, data can be retrieved by querying through the first column without reading all the blocks. Data related to one organization or transaction can be retrieved through the Block Index that increases the efficiency by almost 100% as shown in Figure 11. The search pointer reads only the first column and check for the blocks where drug type is z. The search pointer has no special exploitation in using the Block Index to obtain the transaction history for a batch, as it will have to read all the columns to retrieve the transaction history.

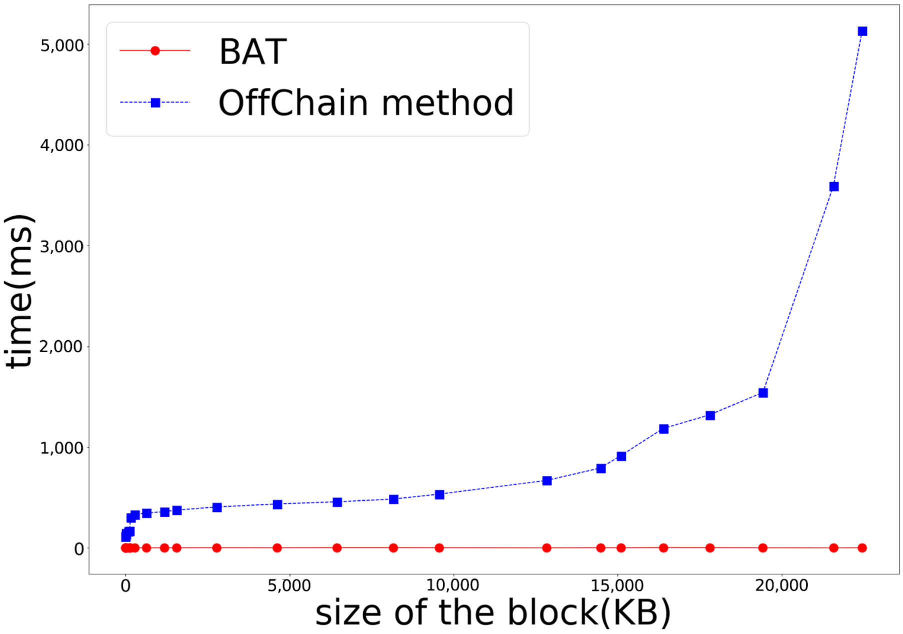


**Figure 9.** The effect of event processing on transaction processing.

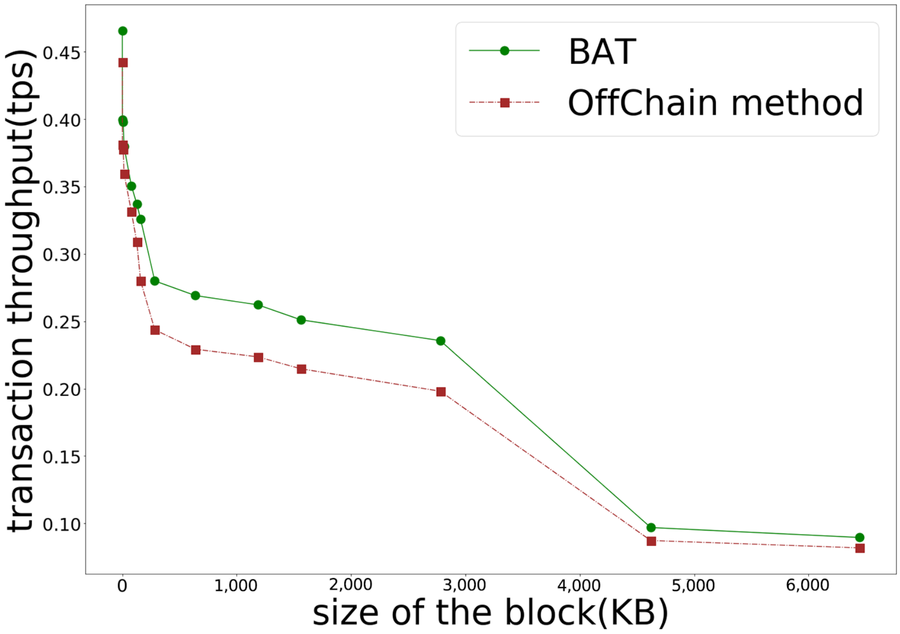


**Figure 10.** Throughput of transactions.

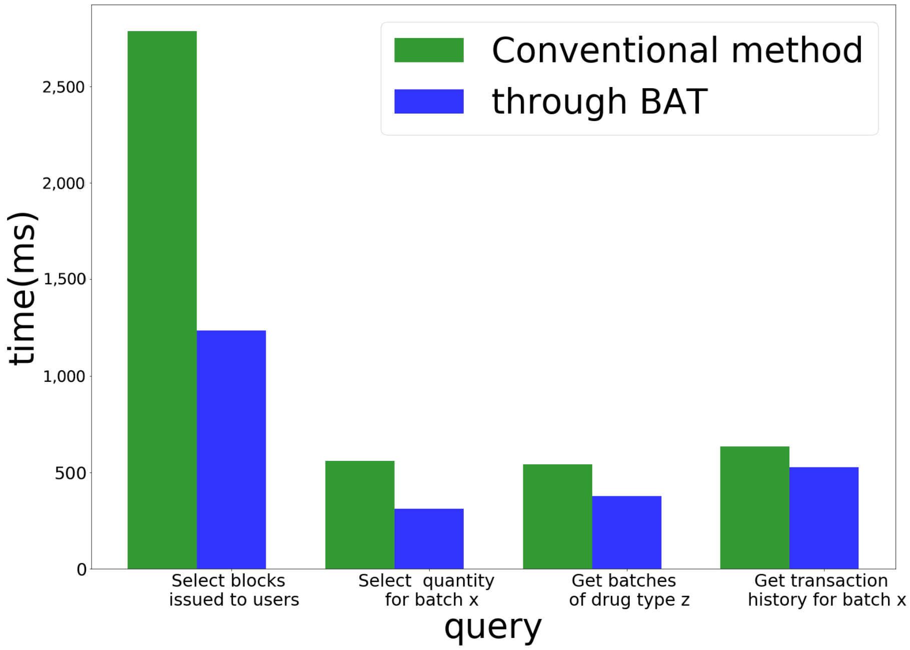


**Figure 11.** Acquiring data from the blockchain.

## 7. Conclusions and Future Work

In this paper, we design and implement a blockchain-based IoT-platform that addresses centralized control, scalability, data security and access control problems found in the current IoT systems. Furthermore, we propose a novel approach called BAT, integrated with the platform that ensures the integrity and authenticity of data used for data-analytics applications. The smart contracts, authentication and access control mechanisms of the platform as well as the BAT ensure about the security and integrity of the data. We present the functionality of the blockchain-based IoT-platform and the integration of BAT with the platform through the use case scenario of the pharmaceutical supply chain.

According to our implementation and evaluation, the novel approach of BAT utilizing BDC saves resources such as storage facilities compared to the conventional approaches of creating a mirror storage that duplicates resources. Furthermore, the total transaction processing time with BAT is considerably low due to the minimal event processing time related to the creation of Block Index compared with the conventional approach of creating off-chain database. The costs associated with data retrieval from the platform are reduced by the usage of Block Index. Currently, batch processing is conducted in the BAT. In the future, we will mainly focus to bring the platform towards edge computing with the optimization of BAT for real-time processing.

**Author Contributions:** Conceptualization, C.E., K.V. and N.B.; methodology, C.E., K.V., N.B.; software, C.E., K.V., N.B.; validation, C.E., J.A., M.S., U.J., N.S., and G.M.L.; writing—original draft preparation, C.E., U.J. and G.M.L.; writing—review and editing, C.E., U.J. and G.M.L.; supervision, J.A., M.S., U.J., N.S. and G.M.L.; project administration, J.A., M.S., U.J.; funding acquisition, G.M.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) [2018-0-00261, GDPR Compliant Personally Identifiable Information Management Technology for IoT Environment].

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horizons* **2015**, *58*, 431–440. [CrossRef]

2. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef] [PubMed]
3. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). In *A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017.
4. Buterin, V. A next-generation smart contract and decentralized application platform. *White Pap.* **2014**, *3*, 37.
5. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *J. Gen. Philos. Sci.* **2008**, *39*, 53–67. [CrossRef]
6. Bahga, A.; Madisetti, V.K. Blockchain Platform for Industrial Internet of Things. *J. Softw. Eng. Appl.* **2016**, *9*, 533–546. [CrossRef]
7. Gramoli, V. From blockchain consensus back to Byzantine consensus. *Future Gener. Comput. Syst.* **2017**. [CrossRef]
8. Nguyen, G.T.; Kim, K. A Survey about Consensus Algorithms Used in Blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128.
9. Novo, O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
10. Huh, S.; Cho, S.; Kim, S. Managing IoT devices using blockchain platform. In Proceedings of the 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, Korea, 19–22 February 2017; pp. 464–467. [CrossRef]
11. Jiang, S.; Cao, J.; McCann, J.A.; Yang, Y.; Liu, Y.; Wang, X.; Deng, Y. Privacy-Preserving and Efficient Multi-Keyword Search over Encrypted Data on Blockchain. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 405–410.
12. Sandhu, R.S.; Samarati, P. Access control: Principle and practice. *IEEE Commun. Mag.* **1994**, *32*, 40–48. [CrossRef]
13. Bhuiyan, M.Z.A.; Zaman, A.; Wang, T.; Wang, G.; Tao, H.; Hassan, M.M. Blockchain and Big Data to Transform the Healthcare. In Proceedings of the International Conference on Data Processing and Applications, ICDPA 2018, Guangzhou, China, 12–14 May 2018; ACM: New York, NY, USA, 2018; pp. 62–68. [CrossRef]
14. Ouaddah, A.; Abou Elkalam, A.; Ait Ouahman, A. FairAccess: A new Blockchain-based access control framework for the Internet of Things. *Secur. Commun. Netw.* **2016**, *9*, 5943–5964. [CrossRef]
15. Hang, L.; Kim, D.H. Design and implementation of an integrated iot blockchain platform for sensing data integrity. *Sensors* **2019**, *19*, 2228. [CrossRef] [PubMed]
16. Zhou, L.; Wang, L.; Sun, Y.; Lv, P. BeeKeeper: A Blockchain-Based IoT System with Secure Storage and Homomorphic Computation. *IEEE Access* **2018**, *6*, 43472–43488. [CrossRef]
17. Gentry, C. Fully Homomorphic Encryption Using Ideal Lattices. In Proceedings of the 41st ACM Symposium on Theory of Computing, STOC '09, Bethesda, MD, USA, 31 May–2 June 2009; ACM: New York, NY, USA, 2009; pp. 169–178. [CrossRef]
18. Xu, Q.; Mi, K.; Aung, M.; Zhu, Y.; Yong, K.L. New Advances in the Internet of Things. *Stud. Comput. Intell.* **2018**, *715*, 119–138. [CrossRef]
19. Popov, S. The tangle. *ABA J.* **2016**, *136*, 1–25.
20. Akcora, C.G.; Kantarcioglu, M.; Gel, Y.R. Blockchain Data Analytics. In Proceedings of the IEEE International Conference on Data Mining, Singapore, 17–20 November 2018; p. 6. [CrossRef]
21. Hyperledger. Hyperledger—Open Source Blockchain Technologies. 2019. Available online: https://www.hyperledger.org/ (accessed on 23 January 2020).
22. Bai, L.; Hu, M.; Liu, M.; Wang, J. BPIIoT: A Light-Weighted Blockchain-Based Platform for Industrial IoT. *IEEE Access* **2019**, *7*, 58381–58393. [CrossRef]
23. Shafagh, H.; Burkhalter, L.; Hithnawi, A.; Duquennoy, S. Towards Blockchain-Based Auditable Storage and Sharing of IoT Data. In Proceedings of the 2017 on Cloud Computing Security Workshop, CCSW '17, Dallas, TX, USA, 3 November 2017; ACM: New York, NY, USA, 2017; pp. 45–50. [CrossRef]
24. Campbell, H. Dangers of Drug Importation: A Case of Counterfeit Cancer Drugs. 2016. Available online: https://catalyst.phrma.org/dangers-of-drug-importation-a-case-of-counterfeit-cancer-drugs (accessed on 22 January 2020).
25. Bayer. Background Information on Counterfeit Drugs. 2014. Available online: https://www.bayer.com/en/background-information-on-counterfeit-drugs.aspx (accessed on 20 January 2020).

26. Rachmania, I.N.; Basri, M.H. Pharmaceutical inventory management issues in hospital supply chains. *Management* **2013**, *3*, 1–5. [CrossRef]
27. Candan, G.; Taskin, M.; Yazgan, H.R. Demand Forecasting In Pharmaceutical Industry Using Neuro-Fuzzy Approach. *J. Mil. Inf. Sci.* **2014**, *2*, 41. [CrossRef]
28. Hunkeler, U.; Truong, H.L.; Stanford-Clark, A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In Proceedings of the 3rd International Conference on Communication Systems Software and Middleware and Workshops, Bangalore, India, 6–10 January 2008; pp. 791–798.
29. Naik, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In Proceedings of the IEEE International Systems Engineering Symposium, Vienna, Austria, 11–13 October 2017; pp. 1–7.
30. Dierks, T.; Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.2. Available online: https://www.hjp.at/doc/rfc/rfc5246.html (accessed on 23 January 2020).
31. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portuga, 23–26 April 2018; pp. 1–15.
32. Thakkar, P.; Nathan, S.; Viswanathan, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In Proceedings of the IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Milwaukee, WI, USA, 25–28 September 2018; pp. 264–276.
33. Devlin, B.; Cote, L.D. *Data Warehouse: From Architecture to Implementation*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1997; p. 432.
34. Galici, R.; Ordile, L.; Marchesi, M.; Pinna, A.; Tonelli, R. Applying the ETL Process to Blockchain Data. Prospect and Findings. *Information* **2020**, *11*, 204. [CrossRef]
35. Dipina Damodaran, B.; Salim, S.; Vargese, S.M. Performance evaluation of MySQL and MongoDB databases. *Int. J. Cybern. Inform. (IJCI)* **2016**, *5*. [CrossRef]
36. Han, J.; Haihong, E.; Le, G.; Du, J. Survey on NoSQL database. In Proceedings of the 6th International Conference on Pervasive Computing and Applications, Port Elizabeth, South Africa, 26–28 October 2011; pp. 363–366.
37. Virtual Machine Instances. Available online: https://cloud.google.com/compute/docs/instances (accessed on 24 February 2020).
38. NHS. Hospital Prescribing Dispensed in the Community | NHSBSA. 2019. Available online: https://www.nhsbsa.nhs.uk/prescription-data/prescribing-data/hospital-prescribing-dispensed-community (accessed on 21 January 2020).
39. Apache. Apache CouchDB-About. 2019. Available online: https://couchdb.apache.org/ (accessed on 21 January 2020).
40. MongoDB. Available online: https://www.mongodb.com/ (accessed on 21 January 2020).
41. Schapire, R.E.; Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.* **1999**, *37*, 297–336. [CrossRef]
42. Huang, X.; Maier, A.; Hornegger, J.; Suykens, J.A. Indefinite kernels in least squares support vector machines and principal component analysis. *Appl. Comput. Harmon. Anal.* **2017**, *43*, 162–172. [CrossRef]