Tangle The Blockchain: Toward IOTA and Blockchain integration for IoT Environment

E. Exposito, Hussein Hellani, Layth Sliman, Motaz Ben Hassine, Abed Ellatif Samhat, Ernesto Expósito, Mourad Kmimech

Tangle The Blockchain: Toward IOTA and Blockchain integration for IoT Environment

Hussein Hellani[1], Layth Sliman[2], Motaz Ben Hassine [4], Abed Ellatif Samhat[3], Ernesto Exposito[1], Mourad Kmimech [4]

[1] Pau University, Pau, France
[2] EFREI Paris, VilleJuif, France
[3] Lebanese University, Beirut, Lebanon
[4] ISIMM Monastir, Tunisia

Abstract. The use of blockchain (BC) technology for IoT-based collaboration platforms is still hindered by its inherent properties, namely, the need for ever-increasing storage and the low scalability. In this paper, we investigate the interoperability requirements between BC and tangle distributed technologies. We propose a new decentralization architecture in which BC and tangle are combined in such a way that the IoT functionality is increased, and storage is enhanced while keeping a high level of reliability, data accessibility, integrity, and security. In our architecture, a BC-based platform installed in the backend, used primarily for data storage and smart contract. In the frontend, the applications are running on a Tangle-based platform so that it fits IoT devices. The architecture, when implemented, will reap the advantages of both technologies while limiting the drawbacks of them. The main part of the proposed approach has been implemented and tested using a GPS data emulator connected to an IOTA node. The received data have been propagated and stored in the deployed BC and IOTA.

Keywords: IoT, DLT, BC, Tangle, Dynamic Ledger.

1 Introduction

Nowadays, IoT is involved in many complex everyday life sceneries [1]. This includes smart vehicles, smart energy grids, smart homes and buildings, and smart portable and wearable devices. To that, we can add more sophisticated or business oriented sceneries such as smart factories and smart supply chains. It is expected to have 25 billion connected devices in 2020, according to the IBM report [2]. From a business point of view, IoT becomes an opportunity at a mass scale. Despite IoT advantages, up to now, no reliable frameworks and infrastructures are designed to connect billions of heterogeneous and disparate devices and their associated services, not to mention data aggregation, data analysis, and decentralized system governance [3].

One good candidate for these issues is BC infrastructure. Actually, BC has many advantages, such as its fully decentralization nature, data integrity, privacy preservation, and anonymity [4]. However, the scalability and transaction cost of these infrastructures are considered today as a real brake for their use in IoT based industry. To cope with this drawback, Tangle protocol, used in the IOTA platform [5], has been designed. Tangle is primarily intended to address scalability issues in a traditional BC-based platform. Unlike the "BC" structure, a "Tangle" consists of a solid mathematical foundation called DAG (Directed Acyclic Graph). Tangle uses a validation process in which transactions are entered into the distributed registry after authenticating two other randomly selected transactions according to a Poisson distribution. Therefore, it is a scalable system and does not require mining or transaction fees. Nevertheless, the level of security provided by IOTA is questionable. In this paper, we establish a comprehensive architecture that integrates BC and IOTA technologies platforms in such a way that we can overcome the scalability issue while providing a high level of security.

On the other side, Ledgers of both BC and tangle are shared between participants and their sizes are in always increasing state whenever transactions are issued and accepted by the network. This adds more challenge to the IoT devices to hold more loads that might be above their capacities in many cases. More specifically, we cannot expect passive users that may audit the ledger intermittently. In that matter, we settle a group of questions concerning the future of these DLT technologies. The main problems are where the data will be stored? How long can users afford data? What is the status of the data in non-stationary cases such as big network failure? And how fort can we rely on independent nodes with non-equal specs and capabilities? All these concerns are related to storage.

The main difference between traditional centralized systems and DLT systems is that database and application are merged in the current decentralized systems. Therefore, storage adds another limitation to the IoT decentralization project. Besides, there are many other drawbacks which vary between DLT technologies. In short, no one decentralized system is free of impediments, and no one alone can fulfill all our business requirements. In this work, we draw attention to some use cases that require interoperability between BC and tangle technologies, and we propose a new decentralization architecture combining BC and tangle.

The contribution of this paper can be summarized as follows:

- The proposed architecture enables the heterogeneous DLTs “BC and tangle” to communicate according to the intra-BC model.
- Decouple database and application (as if in the centralized system) and integrate tangle with BC.
- Introduce the tangle-based dynamic ledger to categorize IoT devices upon their capacities, where we distinguish between three main types of IoT devices.
- Enforce a smart contract to run within the tangle environment via the BC platform.
- The main part of the proposed approach has been implemented and tested using a GPS data emulator connected to an IOTA node. The received data have been propagated and stored in the deployed BC and IOTA nodes.

- Open new research areas for future work to focus on managing applications with two ledgers, in addition to further security layers resulting from two consensus algorithms.

The rest of this paper is organized as followings: In Section 2, we present a brief overview of the related works. The intended use cases and motivation are declared in section 3. We present our proposal architecture in section 4. Validation and implementation are performed in section 5. We settle the benefits and challenges of our proposal in section 6, and finally, we conclude in section 7.

2 Related Works

Our proposal is a combination of inter-BCs solutions, decentralized storage, and a new concept of IoT resource allocations. Thereby, interoperability between BCs and storage would be part of our related work. Sidechain [6], Oneledger [7], Interledger [8], ICON [13], MOAC [14], and Interactive Multiple BC [9] are focusing on integrating different DLT systems explicitly to stream transactions between ledgers. Interplanetary File System (IPFS) [10], Swarm [11], and BigChainDB [12], provide an enhanced decentralized storage solutions. However, these decentralized storage solutions are relying on the participants' storage.

W. Jiang et al. [15] propose a cross-chain interactive decentralized IoT data access model that integrates BC consortium and IOTA tangle to address the IoT scalability and usability. Their provided solution is a combination of BC, tangle, IPFS storage, and notary nodes. IoT devices are mainly working on tangle or sub-tangle platforms; however, IoT devices can work on any BC platform such as Ethereum, Hyperledger, and FISCO BCOS. IoT devices are grouped into sidechains, where each one represents an independent network. BC acts as a controller with a primary role to connect multiple sidechains through notaries. The latter, is a group of nodes that resides between BC and sidechain, act as a gateway for transactions flows between BCs. The notary network confirms each cross-chain transaction by the voting mechanism, and the transaction is approved when the signatures of more than 2/3 notaries in the network are collected. IoT data is stored in IPFS storage, while the BC stores only the hash of these IPFS files. They make use of BigChainDB to address the authentication drawback of IPFS.

This solution provides scalability for IoT devices in comparison to single BC utilization. It is similar to ours in terms of amalgamating BC and tangle, but it differs in both architecture and goals. This proposal offers an explicit integration between the aforementioned DLTs to enable inter-communication. However, we propose the inclusion of tangle-based applications implicitly with a backend BC storage. Besides, our proposal provides BC as a service running on stable cloud nodes, where they are using IPFS storage that is based on the participants' devices.

3 Use Cases and Motivation

Businesses that evolve their BC systems and rely primarily on IoT technology find themselves with a poorly architecture that may affect their business lines. Makhdoom et Al. [16] highlight on difficulties of combining BC with the growing IoT devices. During our study on DLT use cases, we find out some environments that benefit from the distributed system approach, but it explores another drawback of using only one DLT system. A Non-governmental organization collects and sends medications and other humanitarian aids to the war zone. Often, tracing these goods is very important due to the context to avoid that the aides fall in the wrong hands. Furthermore, the connection in such zones is very bad. Besides, nodes have minimal computing and storage resources. Thereby, using BC installed directly on the nodes is not possible. On the other side, installing DLT tangle-based will solve scalability and working offline issues. But a smart contract doesn't run on a tangle-based environment. Also, the peripheral IoT devices are usually incapable of cooperating with BC requirements due to their limited resources and specs. Thus, we can use in this situation our architecture that mix IOTA with BC. The same scenario can be applied for the agriculture-based industry where no robust internet infrastructure and law enforcement is not guaranteed.

4 Proposed Architecture

We propose new decentralization architecture to achieve storage independence, to enhance scalability, and to provide data sustainability and availability at a time.

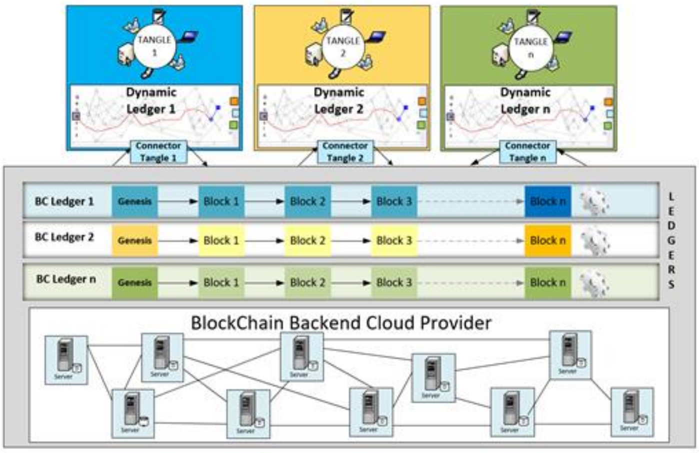

Figure 1. Proposal Overview

The system is composed of the BC backend platform, the tangle-based platform, and the independent connectors that separate the two platforms, as shown in Figure 1. The backend platform is a BC-based system that represents ledgers and data storage. It will be installed in the cloud to provide BC instances for the frontend applications. Ledgers in our backend platform play the role of data storage and backup in the first stage. It has several functions regarding security, privacy, and smart contract in the second stage.

The second main platform is composed of several independent tangle-based applications, distributed along the IoT access-area. Each application has its own tangle-based ledger, namely dynamic ledger. It is dynamic because participants' nodes are free to hold any percentage of the ledger size (detailed in section 4.3). While transactions flow from frontend application towards its final destination within BC ledger, it passes by a specific virtual connector to connect a tangle-based ledger with its BC-based ledger. Transactions are then stored in the BC platform where data is immutable, transparent, secure, and traceable. The connector can communicate with those two ledgers as per our experiment result. From the high-level view, we propose to combine two heterogenic DLT types, integrated each other implicitly, to provide a single end to end solution. In general, users will be dealing with one DLT platform, while the second platform will be embedded within the provided solution.

4.1 Backend BC-based Platform

In our architecture, we employ BC as a service (BaaS) in the backend to store the incoming IoT data instead of surcharging the peripheral devices with high load and increase their responsibilities of being online. However, they still have an option to carry full or portion of their dynamic ledger size, based on their capacities. We settle the ideal BC requirements that fit with the scalable tangle entries. Therefore, BC should be scalable with a fast consensus algorithm with negligible transaction fees [15]. Currently, there is a considerable amount of BC platforms with open sources, that are ready for testing and usage such as Ethereum [17], and Hyperledger [18]. The main challenge of BC performance is the consensus algorithm used to validate the incoming transactions. These algorithms are distinguished by their mechanisms and performance. Practical Byzantine fault tolerance (PBFT) shows a significant improvement over existing consensus algorithms in terms of throughput and latency [15]. PBFT algorithm can perform tens of thousands of transactions per second [19], but it is not scalable with high node entries [20]. However, in our proposed BaaS platform, the number of nodes is logically predefined with high resource capacities in the cloud, which makes PBFT the best choice as a consensus algorithm for our BaaS platform.

As a preliminary step towards decoupling database and application, we observed that the best place for data storage is the public cloud [21] such as Amazon, Azure, etc. However, the application owner is free to use its own private cloud. There are many reasons for "why the cloud is suitable for BC storage". The first thought that comes to mind is that cloud infrastructure is similar to the BC concept in terms of resource distribution. While the "centralized" cloud is found to decentralize the physical data centers to provide high availability, BC is invented to decentralize data itself to ensure transparency. On the other side, with current cloud computing, network latency is the main issue that affects the real-time service for IoT devices. Thereby, running edge computing on top of the BC platform enhances connectivity while reducing the network latency and increase data availability [22].

4.1.1 Structure and challenges

We show in this sub-section how such DLTs integration is achieved. We take into consideration that the backend platform is a fully decentralized solution that supports massive IoT data. Therefore, we settle the criteria of the required storage throughout these three layers, as illustrated in Figure 2:

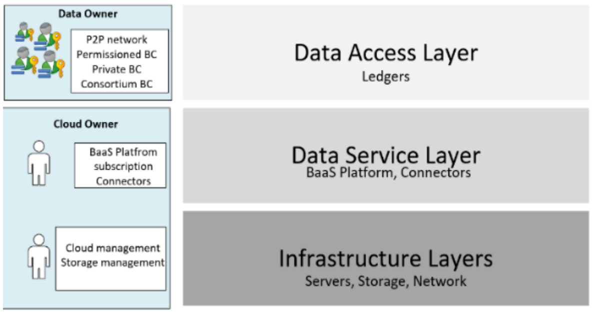

Fig. 2. Cloud structure for DLT enhancement

Infrastructure layer: It is a cloud-based environment with edge computing support and extendible on demand, composed of physical servers and network components on the front side, and storage nodes in the rear side. It represents data centers that are geo-replicated across the world to provide BC service. Components in this layer are managed by the cloud owner, where he is responsible for delivering high service level agreement (SLA). This layer represents the cloud infrastructure service or the physical location of the BC ledger.

Data service layer: represented by the BC platform that runs on top of dedicated servers and storage appliances to provide BC as a service (BaaS) for customers. Every customer has its own subscription. A connector is a part of this layer, replicated along with the BC nodes of the cloud. As shown in figure 2, the cloud owner has access to the physical layer to manage client subscriptions and cloud maintenance.

Data access management layer: represented by the data that are stored within subscriptions. Access to the ledger is concerned by data owners only where cloud owner responsibility is limited to subscription management. Besides, IoT devices or users are referred to their tangle-based access policies that determine their privileges on the stored data in the BC.

The above criteria are related to BC ledger that is in the backend while dynamic ledger is stored locally on each IoT node. The integration of these ledgers is the core of our proposal and experiment.

4.2 Integrate BC with Tangle

Tangle is found to tackle BC scalability issues and high transaction fees [5]. Technically, Tangle is running as a direct acyclic graph (DAG), which constitutes of edge and vertices. A transaction is a vertex, while the edge is the arrow that links two transactions. Transactions are connected in a way that forms the tangle.

Parameters	**Public BC**	**Tangle**	**BaaS + Tangle**
Scalable	No	Yes	Yes
Transaction Fees	Very High	Negligible	Negligible
Require Mining	Yes	No	No
Support work offline	No	Yes	Yes
Fully Decentralized	Yes	No	Yes
Support smart contract	Full	Weak	Full

Table 1. BC vs Tangle vs proposed combination

In table 1, we draw a comparison between tangle, public BC, and the combination of BaaS with the tangle. With tangle, there is no mining nor competition on performing proof of work "POW" to incentivize the winner node. Each node that aims to issue a new transaction has to validate two other ones through inspecting its local ledger content and performs local POW. Also, a tangle-based node can work offline and synchronize their transactions with ledger once connected. On the other hand, the tangle is vulnerable to computing power attacks by 34% domination only [23]. Currently, IOTA is not fully decentralized since it makes use of "coordinator" trusted third party server to fight against computing attack. Another challenge of tangle is smart contract enablement.

BC alone and tangle alone are unable to present a complete decentralized solution. Each one has its advantages and disadvantages. We amalgamate tangle and BC to reap the benefits of them while reducing their limitations to the maximum. The best practice for this new type of integration is to work upon three dimensions: storage, computing, and end-user program. The success of any system will result from the equilibrium of these dimensions, where the weakness of any of these three parameters will impact the whole system.

Storage: In our proposed architecture, the BC ledger is independent of applications and participants. It is cloud-based running as a platform as a service (PaaS) where every data owner will preserve the required space or "quota" to run its application within a fully decentralized environment. With a cloud-based platform, we mitigate the headache behind counting on peripheral devices that have different capacities. Also, with such a PaaS solution, IoT devices and especially light nodes (detailed in section 4.3) are discharged from being up all the time since proposed BC ledger is always-on and ready to serve the online nodes. Accordingly, the tangle-based "dynamic ledger", is the local ledger that is found on every IoT machine where the transaction is created and saved for the first time before being stored permanently in the BC. Since IoT nodes are not equal in terms of resources, dynamic ledger provides the ability to hold a portion of the ledger that fits their capacities. The presence of a considerable amount of IoT nodes maintaining the full ledger is considered a sign for better performance and security.

Computing: it is a sensitive metric in any decentralized system since it reflects time to apply the consensus algorithm and validate transactions. As the frontend tangle-based system is in continuing developments, the transaction confirmation will be shortened, and its validation becomes faster and faster [15]. The BFT algorithm used with the backend BC is up to the computing challenge for certain limit. In future work, we will consider the huge incoming tangle-based transactions and BFT capacity to sort out with the best practice to have maximum throughput.

End-User Program: represented by the software that is shared with the participants themselves or the application of IoT devices. It is a tangle-based platform that includes communication protocols such as MQTT (Message Queuing Telemetry Transport) used in [19]. It permits end users and peripheral devices to interconnect and store their transactions' outputs within a fully decentralized storage platform. The end-user program is one of the three main sensitive metrics that have a direct impact on transaction speed. Thus scalability is affected by the application type and architecture. There are standard measurements and tools (such as average response time, error rate, loop time, etc.) to monitor performance and functionality for any application. In this paper, we did not shed light on the application structure since each one has its own characteristics and protocols. Thereby, our focus is on the combination system design that supports many independent tangle-based applications.

4.3 IoT Resource Allocation Architecture

The IoT device duty is to perform the computation for POW and transaction validation [5], and since they are different in resources and capacities, we distinct in our proposal between three types of nodes: zero nodes, light nodes, and full nodes. Nodes are categorized upon their computation resources related directly to the overall performance.

i. Full nodes are represented by servers with high storage and computing capacities, distributed somewhere near to the edge and hold the entire tangle ledger. ii. We define Zero node, the one which doesn't share its resources with other nodes and makes use of edge computing server or nearest non-zero nodes (if available) to achieve its computation missions. We distinguish between two types of zero nodes, the weak node like IoT device that might have low specs, but are indispensables and essentials in collecting data or doing whatever jobs. This is called permanent zero nodes. The other type called temporary zero nodes that have resources, but it doesn't share those resources with others to fulfill load balancing purposes. Between full and zero nodes, there is a wide range of different light nodes with various capacities that can hold x% of the ledger size, as shown in figure 3. Full and light nodes are non-zero nodes that are engaged in the validation process on behalf of zero nodes. They play a proxy role for zero nodes that load balance the transaction traffic with high throughput. This architecture, when applied, will ensure distributed computing within the dynamic ledger, where loads are somehow distributed equally between tangle nodes based on their capacities. It will enhance the validation procedure and speed up the process of the tangle, especially during peak time.

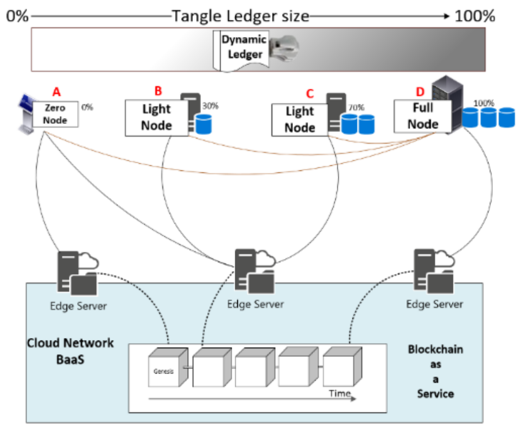

Figure 3. Distributed nodes with different specs

In the example of Figure 3, we present four nodes to distinguish between the IoT devices of the dynamic ledger. Letter 'A' represents a zero node which is looking for available full or light nodes to request the validation process. 'B' and 'C' are both light nodes, however 'C' hold more dynamic ledger size than 'B'. Finally, 'D' is a full node that handles 100% of the ledger. In other words, 'D' has a full copy of the ledger.
Edge servers are found to speed up the transactions process and provide real-time service so that all nodes are connected to their nearest edge servers. Besides their role as a full node, edge servers pass the traffic to the BC ledger via connector with high throughput.

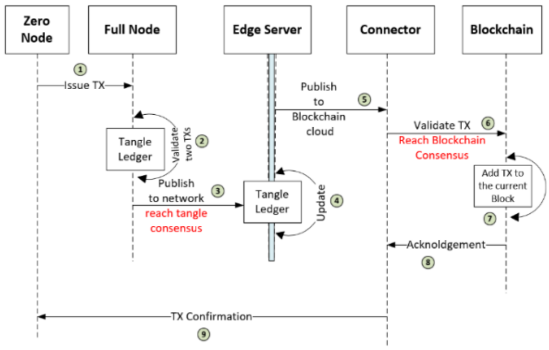

Figure 4. Transaction flow of zero node

In Figure 4, we shed light on zero node behavior as it is the weakest node that requires assistant from other nodes to issue transactions. First, it investigates the nearest available node and issues a request, including all its transaction details. 2) The node that receives the request will inspect the transaction against its local ledger then will validate two previous tips on behalf of the sender (zero nodes). In our example the receiver node is a full node; however, it can be any light node or being the nearest edge server (in case where nodes are not available). 3) The full node performs POW and broadcasts the new transaction to all networks, including the edge server. 4) The nearest edge server updates its ledger similarly to any participant in the system. 5) Edge server pushes new vertex (transaction) to the specific connector. 6) Connector translates vertex to BC form and broadcasts it to all BC nodes to reach consensus. 7) The transaction will be added to the current block after validation. 8) Once the block is closed, the connector receives acknowledgment showing the transaction status. 9) Connector translates back the result to the initiator directly without passing by the validator node.

5 Implementation

To validate the feasibility of our intra-DLT model and to test the interconnection between them, the implementation of both tangle and BC have been realized.

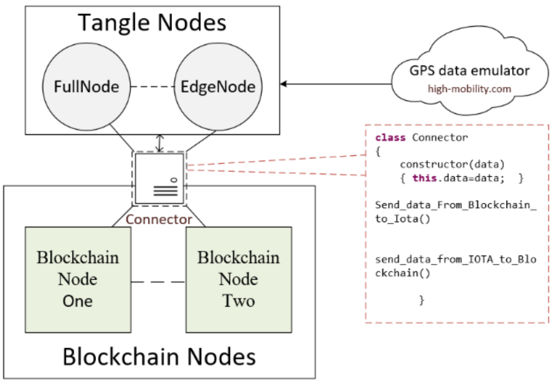

Figure 5. Implementation: Integrate BC with Tangle

Then we validate the transaction flows between tangle and BC. Following our proposal structure, we have created three main components: frontend, Connector, and Backend as shown in Figure 5. i. Frontend: represented by two tangle-based nodes, namely: Fullnode and edgenode. Both Nodes are ".jar" files that run via CMD with the java-jar command. ii. Backend: we have implemented a set of two interconnected nodes of private BC, forming a peer to peer BC network. iii. Connector: This layer is responsible for the link between BC and tangle (IOTA). For our practical case, this part is implemented in Node js (javaScript) as a class named Connector, having a constructor taking the data to be sent from iota to BC or vice Versa.

5.1 Test scenario: send data from IOTA to BC

We have tested a GPS sensor for a virtual car "Porsche Kayenne" issued as an online service at https://high-mobility.com. The data are adapted and stored within tangle nodes, and then it is adjusted to be stored in the BC ledger.

During our test, we executed all the IOTA and BC nodes (FullNode, EdgeNode, BCNodeOne, and Block-chainNodeTow). We launched a program written in Node-js to retrieve GPS data and to adapt it to IOTA. The program sent this data towards the connector in order to pass them to the BC layer. As a result, the issued transactions resided in a JSON file on the BC ledger of both nodes.

6 Results and Benefits

Ideally, the benefits of the proposed architecture include:

- First of all, the database and application are separated. Participants will not rely on each other to get data or to update their database since each application has two different ledgers. Permanent database stored in the BC (cloud BaaS) that is consists of all data starting from genesis until the last transaction. And dynamic ledger, which is tangle-based, exists on each participant node so that it can hold from zero bytes to full ledger size. That is, we have no worries about data storage in terms of availability and physical location since it is a "BC as a service", distributed within big data centers around the world.
- Transactions are flowing from the application towards its final destination inside BC ledger. Hence tangle rules will be applied first. That is, tangle will force BC to follow its footsteps, and as a result, the tangle is chaining BC.
- Enable smart contract on tangle network although DAG has not the concept of time series. The smart contract will be running on the BC platform towards IoT nodes.
- Ability to work offline in some cases, nodes then broadcast their transactions later to the network, thanks to the nature of tangle structure.
- Such a combination eliminates the need for keeping IoT nodes online since storage is a decentralized service and it is always up and running. Extremely, the proposed system supports all IoT devices to be entirely disconnected or being powered off during a specific time.
- Provide resource load balance for different IoT devices where computing tasks are distributed efficiently between them, excluding zero nodes.

7 Conclusion

In this paper, we consider the challenges of the DLT systems in terms of IoT storage and computing. We propose an integration system composed of tangle and BC platforms to undo the merge process of database and application while conserving the decentralization concept. The proposed system will enhance the drawbacks of both systems, such as scalability and storage capacity. Our implementation test shows the ability of such integration. Tangling the BC is a new approach to force BC to run tangle features. Moreover, we differentiate between IoT nodes in terms of resources, and we categorize them upon their capacities. We end up with another proposal for the dynamic ledger that fits with tangle-based IoT to load balance workloads and improve scalability. Future works will focus on measuring the benefits' list and the intermediary connector to mitigate any negative impact of using two ledgers, including a dynamic ledger load balancer. Furthermore, we will consider security sides and measurement of the immunity force against computing attacks with two different consensus algorithms.

References

[1] N. Dey et al. 2018. Internet of Things and Big Data Analytics Toward Next-Generation Intelligence. ISBN 978-3-319-60435-0. Springer.

[2] Pureswaran, V. Brody, P. Device Democracy: Saving the Future of the Internet of Things; IBM Corporation: Armonk, NY, USA, 2015.

[3] Manyika, J. The Internet of Things: Mapping the Value beyond the Hype; McKinsey Global Institute: New York, NY, USA, 2015.

[4] H. Hellani, et al., On Blockchain Technology: Overview of Bitcoin and Future Insights. IMCET, 2018.

[5] Popov, S. The Tangle. Available online: https://iota.org (accessed on 30 Sep 2019).

[6] Back, A., Corallo, M., Dashjr, L., Friedenbach, M., Maxwell, G., Miller, A., Poelstra, A., Timón, J., Wuille, P. Enabling BC innovations with pegged sidechains 2015. Accessed https://www.blockstream.com/sidechains.pdf

[7] D. Cao, B. Binns, E. Zhang, P. Homer, A. Lan. Oneledger, A Universal BC Protocol Enabling Cross-ledger Access through Business Modularization [online] https://oneledger.io/whitepaper/oneledger-whitepaper.en.pdf

[8] Hope-Bailie A, Thomas S. Interledger. Creating a Standard for payments, International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, 2016, 281-282.

[9] Luo Kan et al. A Multiple BCs Architecture on Inter-BC Communication; IEEE International Conference on Software Quality, Reliability and Security Companion, Libson, Portugal, 2018.

[10] J. Benet. IPFS. Content Addressed, Versioned, P2P File System 2015. http://static.benet.ai/t/ipfs.pdf.

[11] Ethereum Swarm 2015 [online] Available at https://forum.ethereum.org/categories/swarm

[12] BigchainDB-GmbH. A BigchainDB primer. [Online] 2015. Available: https://www.bigchaindb.com/whitepaper/bigchaindbprimer.pdf

[13] ICON Officiel community https://icon.foundation/

[14] MOAC Website: www.moac.io

[15] Jiang, Yiming, ChenxuWang, YaweiWang, and Lang Gao. A Cross-Chain Solution to Integrating Multiple BCs for IoT Data Management, 2019. DOI: 10.3390/s19092042 Sensors 19(9):2042.

[16] Makhdoom, I.; Abolhasan, M.; Ni, W. BC for IoT: The Challenges and aWay Forward. In Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, Porto, Portugal.2018

[17] G. Wood. Ethereum: a secure decentralized generalized transaction ledger, 2015.

[18] Hyperledger Fabric. Available online: https://www.hyperledger.org/ (accessed on 01 September 2019)

[19] D. Strugar, et Al. On M2M Micropayments: A Case Study of Electric Autonomous Vehicles. 2018 IEEE International Conference on Internet of Things. https://arxiv.org/pdf/1804.08964.pdf

[20] Han, R., Gramoli, V., & Xu, X. Evaluating BCs for IoT. 9th IFIP International Conference on New Technologies, Mobility, and Security, 2018.

[21] Web site: “BC-as-a-service companies making the dlt more accessible”. https://builtin.com/BC/BC-as-a-service-companies

[22] R. Yang, F.Yu, P.Si, Z.Yang, Y.Zhang. Integrated BC and Edge Computing Systems: A Survey, Some Research Issues, and Challenges. IEEE Communications Surveys & Tutorials, 2019.

[23] Web site:https://builtin.com/BC/BC-as-a-service-companies