Tangle the Blockchain:Towards Connecting Blockchain and DAG

Houssein Hellani, Layth Sliman, Abed Ellatif Samhat, Abed Samhat, E. Exposito

Tangle the Blockchain:Towards Connecting Blockchain and DAG

Houssein HELLANI[a], Layth SLIMAN[b], Abed Ellatif SAMHAT[c], Ernesto EXPOSITO[a]
[a] Universite de Pau et des Pays de l'Adour *E2S UPPA, LIUPPA* Anglet, France
houssein.hellani@univ-pau.fr; ernesto.exposito@univ-pau.fr
[b] EFREI Engineering School-Paris, Villejuif, France.
layth.sliman@efrei.fr
[c] Faculty of Engineering - CRSI, Lebanese University, Hadath, Lebanon.
samhat@ul.edu.lb

***Abstract*—Blockchain brings many added values to modern business systems. However, Blockchain-based applications with massive IoT devices experience some limitations. This is referred to the linear structure and the consensus algorithms used in Blockchain, which consumes the participating nodes' considerable resources. Additionally, IoT devices are generally with limited resources and have limited bandwidth connections. IOTA, which is based on Directed Acyclic Graph (DAG), is a new distributed ledger technology (DLT) for IoT devices. It proves its high scalability by providing parallel data processing. However, DAG is still not mature enough to fully replace Blockchain. In previous work, we proposed combining both Blockchain and IOTA technologies to allow scalable transactions where Blockchain is employed in the backend and Tangle is used in the frontend. In this paper, we consider the proposed solution with the main focusing on the connector part that intermediates both DLT technologies. The connector is a decentralized software component that supports the interaction between the DLT implicitly. The experiments' results show the flexibility to merge both DLTs using message queuing protocol that enables smart contracts to run on the Tangle nodes and enriches the new platform with reliability and working offline features.**

***Index Terms*—Blockchain, IOTA, DAG, Tangle, Connector**

I. Introduction

The distributed ledger technology concept becomes of great importance as it turns into a center of interest in many business sectors [1]. DLT assures to business applications the overcoming of many limitations and the productivity enhancement based on great added values and DLT platform features [2]. It alleviates data traceability gaps, data loss, some centralized platforms' high costs, and enables trust among partners. However, the adoption of Blockchain in IoT system requires significant enhancements [3]. The proliferation of IoT devices in the market requires an advanced Blockchain system to tackle the current scalability issue. The Blockchain scalability is mainly affected by the linear block structure and consensus algorithms. IOTA [4], the alternative DLT platform of Blockchain, is found to tackle IoT's scalability issue, enable micropayments, and resist quantum computing algorithms. Contrarily to Blockchain, IOTA is based on the data acyclic graph (DAG) platform that underlies its ledger named Tangle [5] to adjust with the increasing of the incoming data traffic. The DAG-based architecture enables processing large transactions simultaneously, which nominates IOTA as a scalable DLT system [6] . However, IOTA experiences some limitations as it does not perfectly applies smart contract [7] the same way Blockchain does. Added to this, IOTA requires attaining certain transaction numbers to be considered secure and fully decentralized.

Every DLT has its own protocols that allow its participants to intercommunicate. The main problem is that those protocols are not extended outside their DLT networks and the created environment. Exchanging cryptocurrencies is a good example illustrating this problem with ledgers belonging to independent networks with different protocols and transaction formats. Thus, it is required to flow transactions between different ledgers as if they compose one platform. In a previous work [8] we proposed a solution to combine Blockchain with IOTA, in which the latter represents the front-end application where Blockchain represents the backend platform and a connector part intermediates both DLT technologies. The solution brings many benefits to the industrial market and reduces both Blockchain and IOTA drawbacks. The decentralized applications (Dapps) run over the IOTA platform to achieve scalability so that the massive IoT integration becomes supportable. The incoming traffic is duplicated afterward on the Blockchain backend side to be stored permanently on that ledger. The proposed architecture facilitates IoT integration with the DLT platform on a large scale. It allows running the smart contracts on the Blockchain platform towards IoT nodes, their executable fields. Also, the DAG system offers the IoT nodes the ability to work offline and replicate their data later on. This feature advances such decentralized systems, especially in the complex IoT environments like supply chains. Additionally, this architecture eliminates the need to keep IoT nodes online since the data is always up and running in the backend. In this paper, we consider the proposed solution with a main focus on the connector component. The connector is a decentralized software composed of different javascript libraries based on the message queuing protocol to support the intercommunication between the DLT. It is implemented and tested within the Blockchain and Tangle platforms. The results show its ability to pass smart contracts between the DLTs in an efficient way.

The rest of this paper is organised as follows. We present the related works of the connector in section 2. A brief description of the proposed architecture is given in section 3. We then detail the connector component and its implementation in section 4, and we conclude the paper in section 5.

II. Connector: Related works

Blockchain exists in various types, and they differ in some characteristics such as the transactions' forms, crypto algorithms, and consensus algorithms [9]. The interconnection problem significantly increases due to different networks and organizations running of completely different DLT technology versions and governance rules.

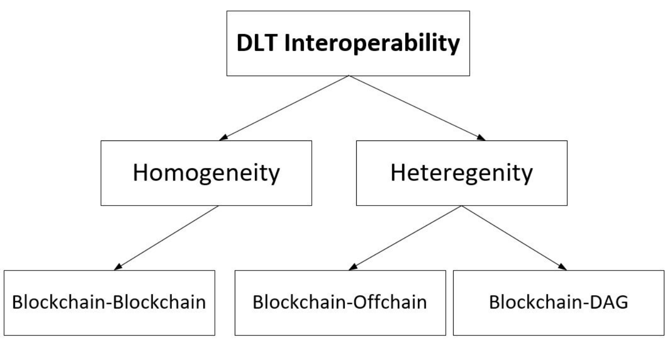

Fig. 1. Interoperability DLT Structure

It resulted in unconnected platforms and siloed from each other, which affect the organizations from reaching their full potential and achieving their goals. Different interoperability solutions are introduced to connect various ledgers and mitigate the gap behind the decentralization concept. There are two main DLT interoperability types, as illustrated in figure 1: homogenous and heterogeneous. The homogenous DLT is formed of either Blockchain or DAG platforms. In the heterogeneous type, the interconnected DLTs belong to different architectures such as Blockchain, off-chain, or DAG.

In the literature, Sidechain [10] is a homogenous solution innovated in 2014 for Blockchains' interoperability and asset transfer facility between different crypocurrencies. It is represented by a second Blockchain connected to the main one through two-way pegged. It is not designed for Blockcahins communication. Instead it processes the transacations circulation within several crypotocurrencies. Another homogenous interoperability solution called Cosmos [11] aims to link Blockchains to each other at a large scale. The Cosmos platform relies heavily on smart contracts, where its architecture is based on the 'hub-and-spoke' system. A series of 'spoke' chains connect to the hub central through inter-Blockchain communication. They build IBC (Inter Blockchain Communication) protocol to communicate between the hub and the other chains linked to the network. Interledger [12] as a homogenous solution is an open-source protocol developed by the W3C Interledger Community Group. It is designed to work within an open network, so interconnection between different parties is facilitated. Its core protocol ILP (interledger protocol), inspired from the internet protocol (IP), is somehow similar to the internet concept. Connectors of interledger work as internet routers where ILP protocol turns all transactions into the same ILP packet size. The main goal of interledger is to remove the barriers between cryptocurrencies and allow payments throughout a predefined network set of connectors. A connector is an explicit participant that has an account on the two different ledgers. Interactive Multiple Blockchain [13] is another proposal to tackle heterogeneous Blockchain integration based on a network of multiple Blockchains called router Blockchain. Any node that joins the network becomes the router of this specific Blockchain system. Routers share a dynamic routing table that is updated whenever a node leaves or enters the network. Since different Blockchains have different transaction formats, a unified cross-chain transaction is introduced for all systems. Two functions are used to achieve the conversion process, pack and unpack. A transaction that is generated by Blockchain "A" will be packed by its router and forward to Blockchain "B To this end, B's router will unpack the transaction and thereby be accepted or denied based on the validation result. The router node transmits transactions according to the routing table written in the router Blockchain. Polkadot [14] is a homogeneous Blockchain network that aims to solve Blockchain extendibility and scalability. It is composed of a relay chain and parachains, where each parachain represents an independent Blockchain. The relay chain is the connector that links these parachains and streams the message between them. With Polkadot, a parachain like "Ethereum" can apply its smart contract to other ones through the relay chain connector. Oneledger [15] is a heterogeneous connector that aims to connect centralized and decentralized applications and remove barriers between these two worlds. It is a gateway for organizations to their decentralized applications composed of API, protocol, and sidechains. Oneledger is called the Blockchain of Blockchains. It provides communication between many independent Blockchains. The consensus is built upon three stages to enable effective integration with these different Blockchain products: business initialization, channel consensus, and public chain consensus. Technically, it consists of a business center, consensus protocol, identity management system, intelligence engine, and Blockchains with sidechains that are attached directly to their core networks. Chainlink [16] is an interoperability solution for heterogeneous DLT to facilitate communications between disparate Blockchain platforms. The resources suppose off-chain data to enable smart contracts and outputs like established payment systems. This solution is vital for Blockchains that do not have to interact with other Blockchain protocols but require external inputs and outputs access. Lastly, The notary approach of cross-chain [17] is designed to fulfill a heterogeneous interconnection between Blockchain and DAG. The notaries, which a group of intersection nodes, play the role of gateways for DLTs of the system. Nevertheless, they do not display the Tangle-Blockchain integration and how notaries connect them.

The solutions listed in the related works enrolled under cross-Blockchain communication [18], where the source Blockchain is the Blockchain in which the transaction is initiated to be executed on a target Blockchain. In our proposed solution, the required connector differs in target and functionality from the above-listed solutions. They target DLT interoperability to allow data transfer between DLTs. In our connector, the source and target DLTs initiate and execute identical records, so the data streams from one DLT to be duplicated on the second. This is to reap the advantages of both DLTs in the first place and mitigate their drawbacks. The idea is that the DAG-based transaction should be stored in the Blockchain with a new format. Hence, a connector with a bridge role is required to satisfy the transactions' flow towards the Blockchain ledger, considering the scalability and security perspectives.

III. Proposed Solution

A. Motivations

Our proposed solution presents the IOTA Tangle architecture as an added-value to Blockchain. Tangle is considered a solution that is mainly designed to address the complications of scalability, transaction fees and mining. Contrarily to Blockchain, the scalability of the Tangle increases when the number of nodes increases. Mining is eliminated totally in the Tangle and replaced by a consensus-building system that pushes the participant nodes to issue a transaction and validate two previous transactions. The issued transaction, including the POW results, must be validated by the next issuer node and so on. Thereby, the consensus mechanism is embedded within the issuer node that performs POW locally. Accordingly, the IoT integration with Tangle is much easier than Blockchain since these weak devices are not charged with mining, transaction fees, and enormous ledger volume (due to the periodic snapshots). Also, the most attractive feature is the parallelism of the data validation, which provides valuable transaction speed. These new enhancements encourage us to avail of the Tangle as a solution that regulates the enormous incoming IoT data rather than mitigating Blockchain drawbacks.

Regardless of its shortcomings, Blockchain is still indispensable because of its accurate timing, full decentralization, and immutable structure. In this proposal, the Blockchain is used primarily to store the data permanently.

B. Combining Blockchain and Tangle

The proposal is composed of three components: the frontend Tangle-based platform, Blockchain in the backend, and a connector that bonds both DLTs, as illustrated in figure 2. Tangle and Blockchain's amalgamation contributes to reinforcing considerable benefits to reducing their limitations at a high level. The combination of two different DLT systems requires an understanding of their different mechanisms where the invented connector is the authorized party to merge both DLTs into one end-to-end DLT platform.

Figure 3 details the mechanism of how a Tangle-based node issues a transaction. In the Blockchainless Tangle, the node should prepare its transaction inside a new bundle (data, recipient address, transaction details, etc..), sign it, select/validate two previous transactions and perform POW locally. Then attach the new bundle of transactions to the main Tangle ledger to be broadcasted to all Tangle nodes. The bundle remains invalidated until some node selects and validates it. A connector that is in listening mode to a new transaction, detects the new transaction, converts it into Blockchain form and deliver it to the Blockchain nodes. Once received, the transaction is subjected to further verification by the Blockchain nodes and being stored permanently in the Blockchain ledger.

1) FrontEnd: Tangle-Based Application: The front-end application runs on top of a Tangle-based platform. Similar to Blockchain Dapps, applications can run separately on the public or private Tangle. The Tangle approach is superior to Blockchain in improving the Iot devices' affinity with the decentralization platform. It empowers nodes to conserve energy since they are not involved in creating blocks and mining. Furthermore, the Tangle considers the various types of IoT devices and categorizes them into full and light nodes. The light node has limited resources should not be directly connected to the IOTA network. Instead, it uses a full node to perform its computing tasks and attach its transactions. Additionally, Tangle empowers the IoT devices to work offline in case of emergency and network disconnection; therefore, the issued transactions must be attached to the Tangle, later on, to be validated by the peers. Allowing nodes to issue offline transactions is necessary for IoT society since most of these devices span many geolocations with unstable network connectivity. These new DLT features motivate IoT technology proliferation and reduce faulty incidents during issuing transactions in a massive IoT environment.

2) BackEnd: Blockchain Platform: Blockchain is employed in the backend of the proposed architecture to store the incoming IoT data permanently. The Blockchain structure allows Tangle users to benefit from its services without being involved in the Blockchain tasks like storing computing, mining, etc. Likewise, the Blockchain nodes are independent of Tangle nodes and the number of IoT devices participating in a Tangle-based application. Therefore, The best practice is to run the Blockchain in the cloud with edge computing

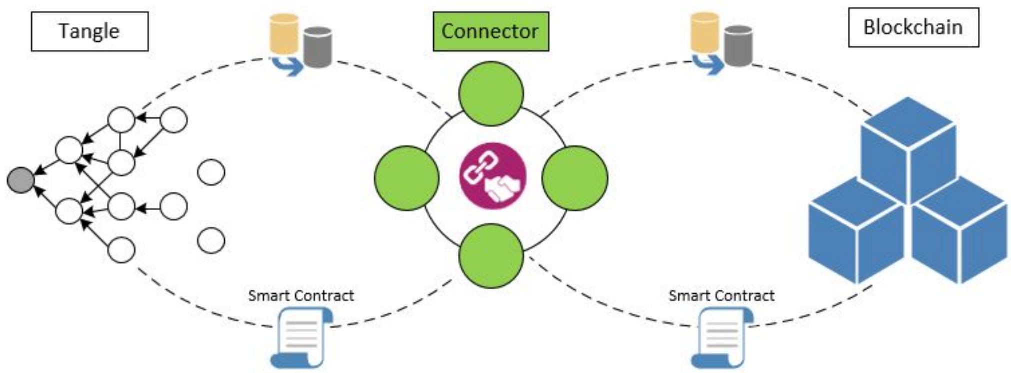

Fig. 2. General Structure

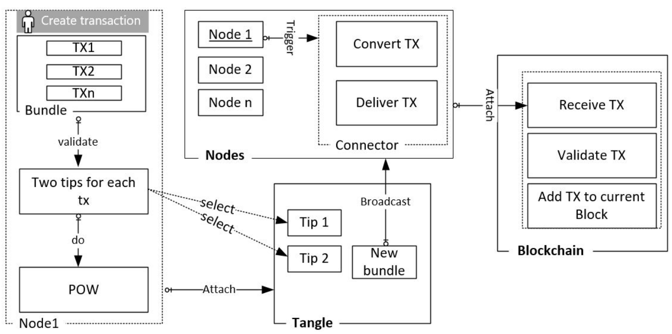

Fig. 3. Proposed Solution

availability to facilitate the connectivity between Tangle nodes and Blockchain [19] , in addition to avoiding latency during transaction propagation. Many Blockchain providers assure "Blockchain as a service" on the cloud with both public and consortium Blockchain service options, such as Microsoft Azure, IBM, Hyperledger Fabric, etc. Technically speaking, any Blockchain type can be integrated with IOTA using the proposed connector. Still, the crucial parameter to be considered in combining Blockchain and IOTA is the number of arrival transactions.

C. Discussion: Benefits and Rate control

Apart from DAG scalability and negligible transaction fees that are suitable for IoT environments, the combination benefits are remarkable. The IoT data spread on two different ledgers, which empowers applications to rely on either Tangle or Blockchain to get the required data. IOTA takes into consideration the weak IoT resource capabilities and performs snapshot every specific time so that not all the data can be found on Tangle. The IoT data are stored on the Tangle ledger and streamed toward Blockchain ledger, the permanent destination. The entire stored in the Blockchain consists of all data starting from genesis until the last transaction. Data storage becomes of no concern in terms of availability and physical location since it is doubled and distributed throughout the Blockchain ledger. Therefore, the application queries the Tangle ledger first for the required data then the Blockchain ledger in case the required data are no longer available on the Tangle. It can be said that the IoT application and its database are partially separated since it has a permanent online full copy in the Blockchain ledger. Furthermore, the latter plays the data backup role for the Tangle nodes in case of corruption or security breaches.

Contrarily to single DLT -either Blockchain or Tangle- users are not relying hardly on each other to execute queries since they have two different data sources, which adds flexibility to the Tangle nodes. In other words, the number of Tangle nodes is independent of Blockchain nodes, and their proliferation does not affect the Blockchain functionality as these nodes are not involved directly in the Blockchain scheme in either mining or block creation. Precisely, Blockchain is only constrained by the number of arrival transactions into its ledger, which could suppose delays in transaction propagation. However, the utilization of an alternative consensus algorithm such as PBFT will mitigate this problem.

Transactions flow from the application site towards its final destination inside the Blockchain ledger. Hence, Tangle rules will be applied first and force Blockchain to follow DAG's footsteps. It seems, as a result, the Tangle is chaining Blockchain. In other words, the features of Tangle and the benefits of Blockhain are still available. The ability to work offline is one of the added values of Tangle, where offline nodes create transactions and broadcast them later to the network. On the Blockchain side, there is a possibility of enabling smart contracts, although DAG has no time series concept. The smart contract will be running on the Blockchain platform towards IoT nodes. Besides, This combination eliminates the need for keeping IoT nodes online since storage is a decentralized up and running service. Significantly, the proposed system supports all IoT devices entirely disconnected or powered off during a specific time.

IV. CONNECTOR COMPONENT

Generally, the software connector is the first-class element of the software architecture [20], defined by interactions between different components' systems. It is categorized under communication, coordination, conversion, and facilities, based on the interacted component services. Each category includes many connector types such as procedure call, event, data access, linkage, stream, arbitrator, adapter, and distributor. Connectors, in general, have different roles: middleware, interaction modeling, architectural styles, and distributed systems. In a heterogeneous system, the connector tends to the adapter type, as it supports interaction between different components that are not designed for interoperation. It involves interaction protocols to match the different parties and mitigate the gap between the different communication channels. The process is done through conversion tasks to synchronize the different parties.

concering the connector types listed above, our proposed connector is classified under adapter type that relies on the conversion process to convert Tangle-based transactions to be readable by the Blockchain platform. The connector is heading to bridge Tangle and Blockchain implicitly. It is logically situated between Tangle and Blockchain ledgers. Physically, it can either reside on Blockchain or Tangle nodes. Also, it can be installed on independent nodes and behaves as distributed connector system. The records of a given Tangle-based application are propagated and replicated to Blockchain ledger. In other words, each record will be stored twice, within the two different DLT ledgers. Transactions validated by Tangle nodes and recorded within the Tangle ledger are immediately forwarded to the nearest connector. The received transaction will be submitted to exact format change to fit with Blockchain architecture.

A. Architecture

There are two main ways to construct the connector: either we agree on a uniform transaction format [13] to be used by both DLTs or build a separate connector that plays the role of translator between DLTs. However, using a uniform transaction format adds many constraints to the system. Due to various application data types, one transaction form will not fit all DLT requirements that could be involved at any given time. Therefore, we adopt translating transactions option, and we use the IOTA javascript package named ”@iota/transaction-converter” to achieve the translation. This package represents the methods used for calculating transaction hashes and converting transaction objects to transaction trytes and vice versa. The Tangle-based transactions are subjected to the IOTA function ”asTransactionObject” to translate them to ”object type” that Blockchain is readable. In the opposite sense, ”asTransactionTrytes” function is employed to translate object transaction of Blockchain directed to the Tangle.

The proposed connector comprises two types of communication channels, zeroMQ (zero message queuing) and Ethereum web3, as illustrated in figure 4. Unlike the communication protocols such as TCP, UDP, and Websocket, the message queuing approach provides processing data in a queue, either in FIFO (first in first out) or according to a priority policy. The main idea is that the data is added to a queue system and executed whenever the caller is ready. ZeroMQ has crucial roles in large-scale distributed systems and enables asynchronous communication [21]. Compared to the single-threaded and multi-threaded queue approaches, ZeroMQ can handle the largest number of users, provides immunity against distributed denial of service attack (DDoS), and scalable [21]. Additionally, it enables working offline with the guarantee that no single record will be lost. This feature enforces the built-in working offline of a DAG-based environment and adds more reliability and flexibility to the system.

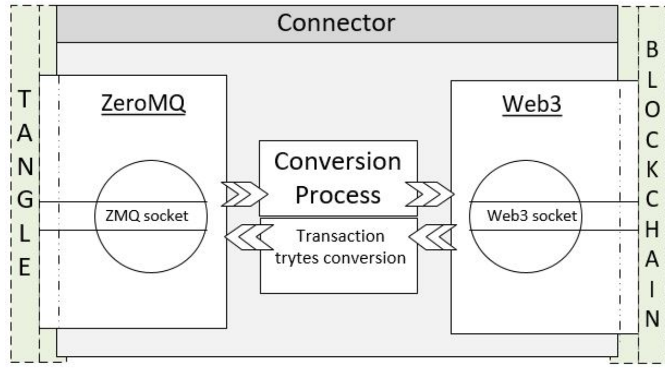

Fig. 4. Connector’s Architecture

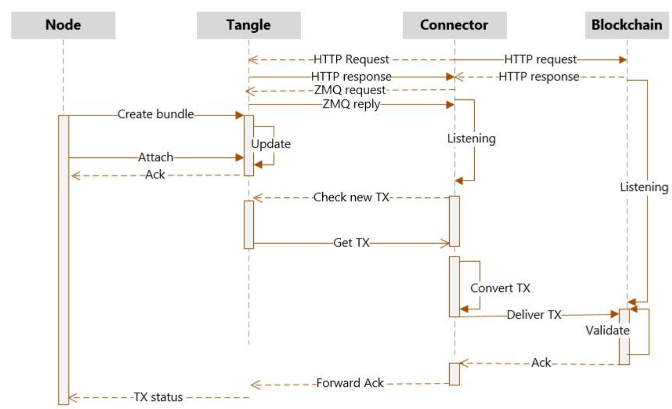

Fig. 5. Proposed sequence diagram

Besides, the Blockchain transactions exploit the ZeroMQ feature of being able to work offline. For example, in a non-stable IoT system, a disconnected node that had triggered a smart contract payment in Blockchain can resume its task when goes back online without losing the initiated transaction.

On the other side, Web3 is a set of communication protocols that allows Blockchain to distribute peer-to-peer transactions without intermediaries. It can interact with Ethereum nodes through HTTP or IPC connections. Using the web3 JavaScript libraries, the connector can interact with smart contracts and retrieve many informational data such as user accounts, send transactions, and more.

Figure 5 details the mechanism of issuing transactions by the Tangle-based node and mirroring them to the Blockchain ledger. At the initial stage, the connector establishes its HTTP connections with both DLTs' nodes. It then establishes a TCP connection with the Tangle nodes for the ZMQ protocol. These connections put the connector in listening mode for both DLTs. The IOTA full node prepares its bundle of transactions (signature, validates two previous transactions, and POW locally) and attaches it to the Tangle. The connector receives the new transaction through the ZMQ protocol and converts it into Blockchain format immediately. Blockchain nodes listening to the web3 connector interface receive the new transaction to be validated and stored in the last block.

The designed connector's primary role is to unify Blockchain and Tangle into one platform suitable for large environments. In addition, it could be used in open collaboration among different DLTs to achieve interoperability. For example, a Dapp Ethereum can communicate with one of the Tangle applications through its connectors. The guest Blockchain benefits from mirroring their transactions with the Tangle application to the main Blockchain. Achieving this kind of interoperability requires adding new connection parameters for the guest DLT similar to the existing ones.

Each Tangle application has its group of connectors, which are independent of the other applications. Connectors are distributed on several nodes alongside the Tangle nodes areas. The connectors run on top of a private Tangle ledger to guarantee its security and transaction propagation speed.

B. Implementation

As illustrated in the figure 6, we deployed three virtual machines to simulate the connector and test its functionality among DLTs. On the Linux-based VM1, two private IOTA instances were installed and configured to share the same Tangle ledger. Each IOTA instance runs through a configuration file (*.ini*) that includes UDP port for Tangle intercommunication, zmq port, and other parameters related to the Tangle structure. The Ethereum Geth node is installed on VM3. The Genesis and all the Blockchain transactions are stored locally on the VM3 disk. On VM2, the connector is installed as a javascript program that includes all required IOTA, ZMQ, and web3 libraries. We create a NodeJS-based application to generate random values towards Tangle 2. Both Tangle 1 and 2 replicate data instantly after being verified and attached to the ledger. The connector is connected to Tangle 1 through HTTP to provide connectivity and through TCP to listen to the data traffic by ZMQ port. On the other side, it is connected to the Geth node through HTTP.

The connector code includes the below different javascript libraries that allow connecting both DLTs:

```
const sender=require('dgram');
const Iota = require('@iota/core');
const Extract = require('@iota/extract-json');
let zmq = require('zeromq');
```

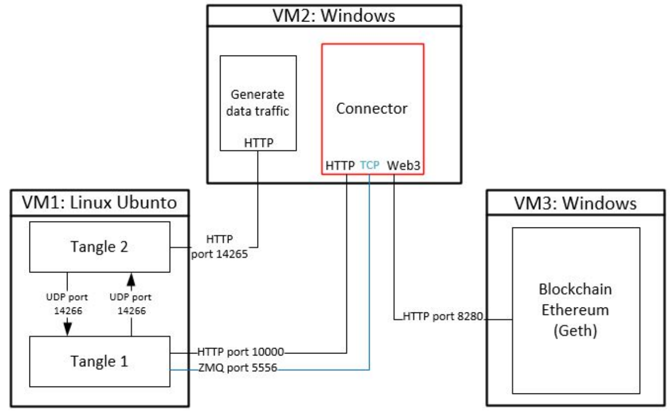

Fig. 6. Implementation bloc scheme

```
let sock = zmq.socket('sub');
const txconverter=
    require('@iota/transaction-converter');
const converter=require('@iota/converter');
const fs =require('fs');
var Web3 = require('web3');
var Promise = require('promise');
```

below is the function used to mirror the transaction in the Blockchain:

```
const txobj =
    txconverter.asTransactionObject(data[]);
const
    writtable_data=JSON.stringify(txobj,null,4);
```

The results of different tests show the flexibility to merge both DLTs using message queuing protocol that enriches the new platform with reliability and enforces the offline working feature.

1) Smart Contract Implementation: We deploy smart contract to realize two main experiments. The first experiment is to validate the scenario when smart contract is created on Blockchain and executed on the Tangle application i.e. the contract propagation from Blockchain node toward Tangle node. In the second experiment, we consider a scenario where Tangle nodes receive temperature values collected from IoT devices and being directed to the Blockchain by the connector. The smart contract's role in this experiment is to detect the temperature which is less than ten or greater than thirty degrees.

The test environment is composed of the Ethereum Remix web with its plugin named Remixd and the Ethereum wallet called Mist, as shown in the figure 7. Remix web is used to develop the smart contract code and interact with the Blockchain. It can work with Web3 objects and can send directly to the "Mist" application. Remix web is typically the best-used application for testing contracts [22]. In a real enviroment the Remix would be replaced by a standard EVM (Ethereum Virtual Machine). In our test, Remixd is installed and activated on the Geth node to be in listening mode through the 65520 port to receive the smart contract and transfer it to the wallet "Mist." Afterward, Remix web application is connected to the local Geth node via Remixd plugin.

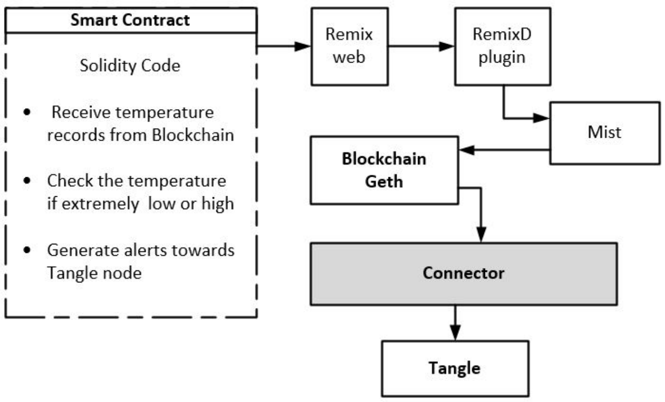

Fig. 7. Smart contract implementation

To fulfill the first test and validate the smart contract propagation towards tangle, we create a contract with simple function that return "hello message". the contract is deployed through Mist and stored on the Geth node. The connector, which is in listening mode, detects the new contract, converts it into trytes, and delivers it to the Tangle through zmq port. Thus, the Tangle gets the smart contract message and register it within its ledger. In the second test, we deploy the smart contract represented in figure 7, on the Remix web to control the temperature values received periodically by the Tangle. Once enabled, the contract checks every value and triggered if the detected one is below ten or more than thirty so that it alerts that something wrong happens. This incident passes to the Blockchain ledger in the first stage to be detected by the connector immediately, exposed to change in its format, and then delivered to the Tangle. The connector includes this function that allows it to push smart contract to the Tangle:

```
const txobj =
    txconverter.asTransactionTrytes(data[]);
```

These experiments show the ability to run Blockchain' smart contract on the DAG-based application through the intermediary connector.

V. Conclusion

This paper presents a solution to facilitate the integration of massive IoT devices with decentralization technology. The proposed solution combines Blockchain and IOTA technologies into one platform where the decentralized applications can be installed in the Tangle, the frontend component. In the backend, the Blockchain is installed to mirroring the data and storing them permanently. In the middleware, a connector is responsible for managing the Tangle traffic towards Blockchain and vice versa. We highlight the connector component and detail its architecture as well as the implementation. The experiments show that the connector's efficiency in connecting both DLTs allows us to integrate a massive number of IoT devices and enable the smart contract on the Tangle side. Also, we reap many other features such as high scalability, the ability to work offline, and low transaction fees. In future work, we will exert additional enhancement related to the connector interoperability, and we will extend the utilization of smart contracts to be triggered on the DAG-based software with Application Interface while executed on the Blockchain.

References

[1] H. F. Atlam, A. Alenezi, M. O. Alassafi, G. Wills, Blockchain with internet of things: Benefits, challenges, and future directions, International Journal of Intelligent Systems and Applications 10 (6) (2018) 40–48.

[2] H. Hellani, A. E. Samhat, M. Chamoun, H. El Ghor, A. Serhrouchni, On blockchain technology: Overview of bitcoin and future insights, in: 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), IEEE, 2018, pp. 1–8.

[3] A. Rejeb, J. G. Keogh, H. Treiblmaier, Leveraging the internet of things and blockchain technology in supply chain management, Future Internet 11 (7) (2019) 161.

[4] R. Alexander, IOTA - Introduction to the Tangle Technology: Everything You Need to Know about the Revolutionary Blockchain Alternative, Independently published, ISBN:198041887X 2018.

[5] S. Popov, The tangle, White paper 1 (2018) 3.

[6] C. Fan, H. Khazaei, Y. Chen, P. Musilek, Towards a scalable dag-based distributed ledger for smart communities, in: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Ieee, 2019, pp. 177–182.

[7] M. Almakhour, L. Sliman, A. E. Samhat, A. Mellouk, Verification of smart contracts: A survey, Pervasive and Mobile Computing (2020) 101227.

[8] H. Hellani, L. Sliman, M. Ben Hassine, A. E. Samhat, E. Exposito, M. Kmimech, Tangle the blockchain: Toward iota and blockchain integration for iot environment, Springer 2019 HIS India.

[9] T. Koens, E. Poll, Assessing interoperability solutions for distributed ledgers, Pervasive and Mobile Computing 59 (2019) 101079.

[10] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, P. Wuille, Enabling blockchain innovations with pegged sidechains, URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains 72.

[11] B. E. Kwon, J., Cosmos whitepaper, Whitepaper. https://github.com/cosmos/ cosmos/blob/master/WHITEPAPER.md.

[12] S. Thomas, E. Schwartz, A protocol for interledger payments, URL https://interledger.org/interledger.pdf.

[13] L. Kan, Y. Wei, A. H. Muhammad, W. Siyuan, G. Linchao, H. Kai, A multiple blockchains architecture on inter-blockchain communication, in: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE, 2018, pp. 139–145.

[14] G. Wood, Polkadot: Vision for a heterogeneous multi-chain framework, White Paper.

[15] Oneledger, Oneledger white paper, URL https://www.oneledger.io/hubfs/Website/Whitepaper/oneledger-whitepaper.en.pdf.

[16] M. Kaleem, W. Shi, Demystifying pythia: A survey of chainlink oracles usage on ethereum, arXiv preprint arXiv:2101.06781.

[17] Y. Jiang, C. Wang, Y. Wang, L. Gao, A cross-chain solution to integrating multiple blockchains for iot data management, Sensors 19 (9) (2019) 2042.

[18] R. Belchior, A. Vasconcelos, S. Guerreiro, M. Correia, A survey on blockchain interoperability: Past, present, and future trends, arXiv preprint arXiv:2005.14282.

[19] R. Yang, F. R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: A survey, some research issues and challenges, IEEE Communications Surveys & Tutorials 21 (2) (2019) 1508–1532.

[20] M. Shaw, Procedure calls are the assembly language of software interconnection: Connectors deserve first-class status, in: Workshop on Studies of Software Design, Springer, 1993, pp. 17–32.

[21] F. Akgul, ZeroMQ, Packt Publishing Ltd, 2013.

[22] A. M. Antonopoulos, G. Wood, Mastering ethereum: building smart contracts and dapps, O'reilly Media, 2018.