Modular Architecture for Refrigerant Traceability in Supply Chains using Ethereum Blockchain

Brendan McArdle
School of Electronic Engineering
Dublin City University
Dublin, Ireland
brendan.mcardle6@mail.dcu.ie

***Abstract*—The ability to trace product origin and movement is key to supply-chain quality control in manufacturing. This time-consuming mainly paper-based process is not only complicated but also error-prone. It is also open to fraud as the data can be easily modified. The widespread connection of devices has created new opportunities to bring efficiencies to the manufacturing supply-chain through automation of data collection and also business processes. Moving information from paper to digital allows for easier collection, exchange and storage amongst other benefits. In this paper we present REFTOB - (Refrigerant Tracker on Blockchain), a decentralized IoT solution for tracking delivery and storage of refrigerant, allowing for easy traceability right from the supplier to the transporter, onto the industrial customer. We evaluate the performance of the blockchain solution through testbed experiments and show that, where the volume of data is controlled, both local and hosted Blockchain solutions outperform the manual process.**

Keywords—Internet of Things, IoT, Blockchain, MQTT, Smart Contract

I. INTRODUCTION

The ability to trace materials as they pass through the supply chain is a common requirement for manufacturers. This is even more critical when it comes to a product like refrigerant gases (Fluorinated or 'F-gases'), where companies are mandated under EU law [1] to account for every kilogramme of the product purchased and consumed during the manufacturing process. An inability to track products effectively means companies can incur compliance fines [2]. Sharing this product information across multiple parties brings its own challenges that have not fully addressed in 'standard' technologies to date. Many companies that rely on paper face daily challenges ensuring the data is collated correctly. Where paper records are kept for multiple sites daily this quickly becomes unwieldly. Client-server software applications exist to manage this data, but these are installed or owned by one of the parties. Where trust issues occur between parties (did the other party alter a document?) a centralised application has certain limitations. It falls short of instilling confidence with all of the parties that the data has not been modified.

A decentralised solution where each party keeps a replica of the entire data history would address these trust concerns. Blockchain has emerged as a new technology that could be used to implement traceability by creating the data trail while also maintaining data integrity and security. Each party (or member) of the network maintains their own node containing an identical copy to the other nodes. Transactions recorded on one blockchain must be identical across all blockchain nodes. This distributed leger of transactions has other benefits like redundancy and smart contracts that make it worthy of investigation for the challenges outlined.

The Internet of Things (IoT) [3] is a network of sensing devices with limited resources and capable of wired/wireless communications with cloud services and designed to handle sensitive data with security and privacy key to the overall architecture. With addition of IoT to the solution there is now a manageable means of collecting much of the required data at source. Messaging protocols form the link between these different devices and are key in any solution design.

In this paper, we design and implement a solution for capturing field and factory data, transmitting the data to remote servers, where it is stored on an Enterprise-grade Ethereum blockchain. The motivation behind this paper is to improve the way data is exchanged between multiple parties when it comes to tracking these GWP (Global Warming Potential) F-gases. For centuries, businesses have trusted a signed piece of paper - the delivery docket. The problem with this age-old approach is fraud and inconsistency - as well as the speed of sharing data. The solution proposed here aims to replace paper-based systems with a decentralised architecture that is transparent, trustworthy and can be accessed instantaneously by a network (consortium) of businesses that are part of a supply chain. Furthermore, collecting data from connected devices, exchanging this data, managing access, and ensuring the underlying immutability - these are elements critical to traceability and inherent to the solution.

Using the novel approach presented in this paper, we illustrate how IoT and Blockchain can be used to automate a slow paper-based manual process and allow data to be exchanged easily between parties where trust is not guaranteed. The focus is on the data collection and transmission process and how this interacts with cloud-based elements. The solution makes use of Open-Source Software (OSS) to leverage existing work. Where this occurs, it is outlined in the text and included in code comments.

The remainder of this paper is structured as follows: Section II describes related work using IoT infrastructure with blockchain; Section III shows an overview of the proposed solution and description of the modules and how the interconnect with each other; Section IV discusses the results from our experiments and any limitations encountered during the testing process. Section V concludes the paper and outlines possible future work in this area.

II. BACKGROUND AND RELATED WORK

Integrations of blockchain and supply-chain receive widespread attention, which is understandable as the features and properties of traceability, transparency and reliability that are inherent to blockchain readily enhance the supply-chain.

Arena [4] proposes a blockchain-based application for the traceability and certification of Extra Virgin Olive Oil. This solution focuses on a dynamic auto-tuning of blockchain parameters to ensure timely publishing to the blockchain. It is conducted on a Hyperledger Fabric blockchain with a very high transaction rate (100 transactions/s). Our proposal evaluates Blockchain-as-a-Service (hosted by Kaleido [5]), which appeals to large corporations, removing the onus of performance to the provider rather than the business consortium. The proposed transaction rate in our test environment is much lower (1 t/s to 10 t/s) as is involves transported goods being weighed once every one to ten minutes. Cui [6] propose a Hyperledger Fabric solution for the electronics component supply-chain focussing on a physical scanned component and its associated product data. Tian [7] examines a food traceability solution using blockchain relying on RFID tags. Hongyan Cui [8] propose a Blockchain-based solution but using CoAP messages with management hub for devices. Our solution relies on MQTT as a lightweight messaging protocol. Sun and Ansari [8] look at mobile connected devices and self-management during different stages of their life cycle. Privacy and Control of the IoT devices is critical to data security, however in our proposal this is managed by AWS IoT Core using device certificates and adequately meets the requirements. Teslya [9] gives a comprehensive summary of blockchain platforms being used in Industrial IoT, however stop short of assessing commercial offerings, preferring to focus on the questions to be asked when selecting a Blockchain-as-a-Service provider.

The objective of this paper is to look at the modular aspect to Blockchain-as-a-Service (BaaS) and how this integrates with other existing cloud-based *function*-as-a-service [10]. This work looks at the integration of these emerging technologies, as it is often the barrier to widespread adoption in industry and commonly the first hurdle where project implementation meets reality and flounders. It adopts a simple micro-controller (RaspberryPi) for the device connectivity in the field.

Modern manufacturing requires a platform approach to architecture where elements can be switched in or out as they evolve and become optimised over time. Additionally, this paper assesses the ability of *permissioned [11]* Enterprise-grade blockchain to contribute to this rapidly changing environment. Smart Contracts are key to this platform approach, where off-chain processes can be interrogated or activated. The novel aspect of the solution proposed is the modular architecture which ensures flexibility for future improvements in any element within the system.

While the benefits are clear, there are obvious challenges [12] with the evolution from paper-based systems to digital - and still maintain trust. This solution comes with some challenges which were incorporated into the design process**:**

- Security and privacy - devices can only publish data based on access controls which need to be managed correctly.
- Scalability - whilst the design proposed here is tested on a smaller number of devices, the design philosophy is for a scalable enterprise solution. The Blockchain Trilemma (Decentralized, Scalable and Secure) is well documented [13] and core to any proposed solution.
- Performance - digitizing any business process must not come at the cost of speed. The system is assessed in terms of time to execute
- Cost - the target consortium includes a mix of customers such as small to medium sized transporters working alongside large refrigeration suppliers and multi-national equipment manufacturers. This heterogeneous environment must accommodate all parts of that digital maturity base.
- Support – the resulting solution must be easy to support remotely

III. SYSTEM MODEL

This section presents the REFTOB (Refrigerated Tracker on Blockchain) system as a blockchain-based IoT system for product traceability related to refrigerant delivery and storage. We begin with a description of steps that comprise the current business activities. This is the backdrop against which our proposed solution will enhance.

A. Business Activities

The REFTOB system is designed to optimize an existing business process. The established business activity is defined briefly as follows and outline in Figure 1. An order is placed by the customer (a manufacturing company) for refrigerant based on demand generated by the factory operations.

1) The refrigerant supplier fills a delivery truck with R-452A refrigerant, and the truck is weighed exiting the supplier's location on its way to the ferry.
2) Truck travels to a weighbridge (weighing scales - field device connected) nearest the customer.

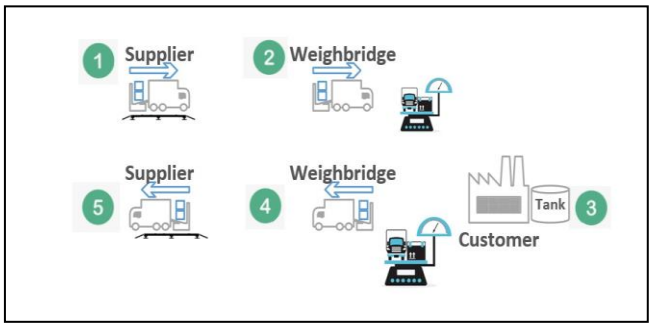

Figure 1 Business Process map

3) Delivers refrigerant to the Customer's bulk storage tank (factory floor device - PLC connection) at the factory (for example, 25,000 kgs of R-452)
4) Truck returns to the weighbridge to be weighed
5) The truck returns to the Supplier and the truck is weighed entering the site.

One problem this paper attempts to address is the following: data from step 2 is recorded on a printed docket (date, customer) with the recorded pre-delivery weight added by hand and signed out by the truck driver. When the truck returns in step 4 the same piece of paper has the (reduced) post-delivery weight entered by hand. This cumbersome process is further slowed down by the obligation to scan this delivery docket and share it with the other stakeholder – supplier and customer.

Together these three entities (Supplier, WeighBridge and Customer) form the members of the REFTOB blockchain network. The process of adding members to the consortium, managing their key access, commissioning their respective infrastructure nodes is managed by the Kaleido BaaS platform.

Table 1 shows the proposed modular framework of REFTOB with its three core modules. The Solidity element in Module D was tested on both local and cloud-hosted blockchain implementations.

TABLE I SYSTEM ELEMENTS

Module	Element	Location	Activity
A	Field sensor	Field	Collect
A	RaspberryPI	Field	Connect
A	Storage sensor	PLC	Publish
B	Node- Server	EdgeServer	Transmit
B	BC_NodeServer	EdgeServer	Transmit
C	Blockchain	Kaleido	Store
C	MQTT Broker	AWS IoT	Broker
D	Solidity	Blockchain	Logic

In Module A, IoT technologies are used to collect and transmit data both from the field and the factory floor. Subsequently, the application integration module ensures the exchange of the data from the cloud-based MQTT broker to the Edge Servers for transmission to the blockchain, in Module B. The connections between Modules A, B and C can be clearly seen in Figure 2. The design includes Edge Servers to allow local testing in a simulated blockchain environment (Ganache) but which also facilitates pushing the data to hosting provider Kaleido running a private Ethereum chain. Module C illustrates the scalability aspect of the solution where Enterprise-grade infrastructure is required. Figure 2 illustrates how the different modules interact with one another.

B. IoT Module - Data Collection and Transmission

While we chose a temperature sensor (*DS18B20*) as a simplified data source for illustration purposes in this paper, in reality this would be a load cell attached to an industrial scale at the weigh-bridge.

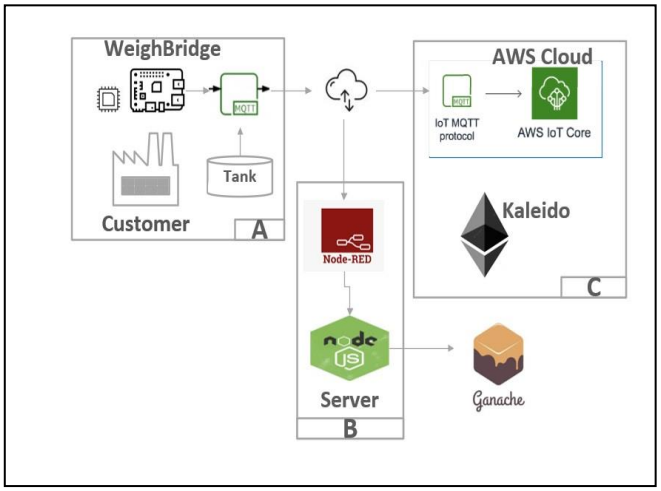

Figure 2 REFTOB Modular Architecture

This client is attached to a RaspberryPi 4.0 which publishes the data via MQTT to the AWS IoT Broker via Node-Red running locally to the micro-controller. The second client is a factory floor based PLC (Programmable Logic Controller) which stores data in a Tag register, connected to the SCADA (Supervisory Control and data acquisition) platform, which is Ignition by Inductive Automation. This is a common SCADA platform in mid to large sized manufacturing companies and was selected to mirror a real-world manufacturing architecture. Figure 3 shows how the SCADA layer connects to the PLC over OPC (Open Platform Communications), reads the data tag value (for the on-site factory Bulk Storage Tank) and publishes the corresponding JSON data block to the MQTT Broker in AWS IoT, which also includes an accurate timestamp which is used during the test phase to measure performance. MQTT was chosen as a messaging protocol as it is lightweight and not payload

```
def valueChanged(tag, tagPath, previousValue, currentValue, initialChar
    val = system.tag.readBlocking("[.]Ref_Tank_Weight")
    system.cirruslink.engine.publish("mqttAWS_US","Brendan/Tank_Open",s
```

Figure 3 SCADA Platform publish MQTT to AWS IoT

prescriptive. Primarily designed for small, connected devices, it has a low overhead and small message size. Because of its asynchronous communication model it is particularly suited to scenarios where connectivity could be intermittent, e.g. in the field.

Publishers are often the sensing devices and subscribers are actuators (in our design the subscribers are the Application Layer, in effect, software actuators). IoT Communications protocols like MQTT can support secure channels via DTLS.

The data flow between Module A and Module C is intended to show that different data sources can be aggregated in the cloud and data analytics applied to them. The solution

reads a data value change event from an Industrial PLC and publishes the data to AWS IoT Core. The Ignition SCADA platform has access to the MQTT Broker in AWS IoT Core by virtue of keys loaded onto the on-premises Ignition server. These device certificates are generated on the AWS IoT platform and ensure MQTT messages are transmitted from the SCADA platform to AWS IoT.

C. Application Integration Module - Edge Servers

This layer connects the various pieces of the system together and ensures the data is exchanged securely between the different elements in the system. The node-red server subscribes to the data topic in the cloud and consequently can trigger an update to the blockchain with any fresh data changes. It passes this data to the BC_NodeServer in the form of a http request (URL + query string). The BC_NodeServer acts as an interface to the Ethereum blockchain. By simply changing a configuration file on the BC_NodeServer we can switch from our local test environment (Ganache) to an Enterprise-grade Blockchain-as-a-Service environment when production ready. There is a very basic UI which publishes a HTTP Response string to the browser simply to indicate a value change. A comprehensive UI application was not part of the requirements for the design process.

D. BaaS Modules - Data Storage and Data Brokerage

The data from the devices is published to a MQTT broker hosted in AWS IoT Core, which ensures only certain data sources can be published to this broker, using public key encryption. The device data is written to the blockchain via BC_NodeServer - the (Blockchain as a Service - BaaS).

Ganache is a local blockchain that can be run on Windows Desktop and is part of the Truffle Suite ecosystem. It allows developers to commission a personal Ethereum blockchain on which they can test Solidity smart contracts. Once unit-tested, these same contracts can be deployed to a Blockchain-as-a-Service provider such as cloud-based Kaleido which has scalability and security built in as an off-the-shelf offering.

Ethereum is an open source decentralized blockchain platform, built upon technology like peer-to-peer networking. Ethereum was chosen as the target blockchain as it has a rich ecosystem of developer tools mainly due to the maturity of the protocol. The selected hosting provider (Kaleido) has three Enterprise offerings when it comes to selecting a protocol for a Blockchain : Ethereum, HyperLedger Fabric and Corda. Private Ethereum was chosen for this solution as it offers the security and privacy required by the business consortium. Publishing refrigerant consumption data openly on the internet could be exposing commercially sensitive information. Kaleido has a hardened Go implementation of the core Ethereum node which utilizes a Proof of Authority (PoA) Consensus [14].

E. Solidity Smart Contracts

Smart Contracts are programs that are written and compiled into the blockchain in digital form. A contract comprises a collection of code (its functions) and data (its state) that resides at a specific address known as the contract address.

Smart Contracts are written in Solidity programming language, compiled by the Ethereum Virtual Machine into bytecode and executed on the blockchain. Smart Contracts are especially useful in a business consortium, where multiple parties need to transact and also execute business processes where there is no trusted central authority. These transactions between members in the consortium are verifiable, immutable and securely distributed across the network, giving members full visibility and ownership of the transactional data. Smart

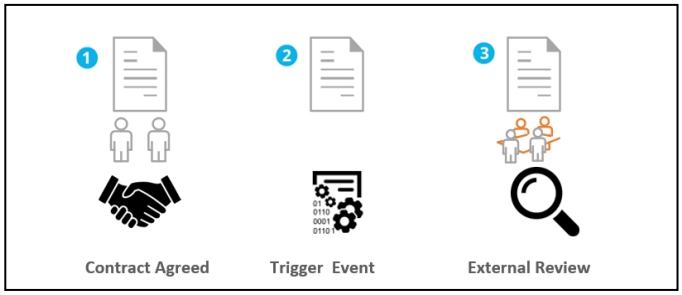

Figure 3 Ethereum SMART Contract

Contracts are replicated and distributed over all of the nodes of a blockchain network. The Smart Contract runs in a VM (Virtual Machine) on the blockchain, which has no direct connection to external networks.

A key motivation behind this paper was to add business process automation in order to fully exploit the IoT data sources and demonstrate this time-saving capability. In doing so, it also illustrates how errors are removed from the system. In our basic example, the Order to Supplier could be automatically placed once BulkTank goes below a certain level. The SmartContract *WeighBridge.sol* is tested on Ganache before deployment to Kaleido in Module C. Kaleido provides the integration opportunities that would execute an email *Send* request which are considered in future work explored at the end of this paper.

F. Open Source Software

The solution makes use of open-source software to provide frameworks for modular code. The Truffle Suite from Consensys provides the initial framework with Microsoft's VSCode plugin to create contracts and attach to a Ganache blockchain locally and the Kaleido blockchain-as-a-service remotely. Boilerplate code (Truffle Box) from Kaleido designed specifically for the Truffle development environment provides guardrails on how to connect to Kaleido [15]. The BC_Server is built using Nodemon which in turn is built on node.js [16]. Both the local sensor device in Module A and the edge server in Module B make use of Node-red, a browser-based flow editor that renders it a simple task to wire together different nodes in a project.

IV. Experimental Results

To investigate the performance of our modular architecture we measure the efficiency of the field devices saving data to both local and cloud-based blockchain platforms - the testing measures the time to write to the blockchain. This is compared with the current benchmarked manual process which takes on average 3-5 minutes for the weighbridge operator to weigh the truck, sign the docket and

scan it into an email, simultaneously sharing it with the customer and supplier. This manual process ensures all parties to the transaction – the supplier, the transporter and the customer – have access to the data in the event of a disagreement or query. The comparable activity in this proposed solution is the recording of the weight values on the blockchain.

A. *Definition of the testbed environment*

The field device is a DS18B20 temperature sensor transmitting data to the cloud-based MQTT broker via a RaspberryPi v4.0 (Broadcom BCM2711 SoC with a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor). The Siemens S700 Industrial PLC is connected over ethernet to the AWS IoT Core MQTT broker. The data stream is simulated using a temperature sensor as the planned load cell proved to be of unreliable quality during the experimental phase. In the real scenario this would be an industrial sized scale at the weigh-bridge site with a digital output. The SCADA platform is Ignition v8.1 from Inductive Automation running on an on-premises Windows Server, close to the factory operations. For local writes to blockchain the Ganache application is running on a Windows 10 Desktop PC running on Intel i5. For scalability testing the Kaleido Enterprise platform offers Blockchain-as-a-Service running on AWS/Azure or Google Cloud. The BaaS environment in the test environment is based on the StarterPack and has 1Gb of memory and 0.5 vCPU. By increasing the spend this environment can be scaled up to 4Gb of memory and 2 vCPU. The AWS data centre is located in US-East1 zone which would have implications for latency, this is not an issue in this solution were data is being recorded every few minutes, rather than every few milliseconds.

B. *Experimental Results*

For the proposed scenario, four test cases have been carried out. A test consists of a field device recording data, and it being stored on the Kaleido blockchain. Our use case indicates one message every 10 seconds is acceptable. However, through stress testing by increasing message speed we want to determine any limitation.

Field device to Blockchain: this records the time the data block travelled from the field device, via the RaspberryPi to the MQTT Broker, and ultimately when it was written to the hosted Kaleido blockchain. The field device was chosen over the PLC, as we have no means of modifying the speed of incoming data from a production PLC.

Table II shows the results for 0.1 messages per second (msg/s), which equates to one every 10 seconds. The delay from read to write is consistently 5 seconds and indicates one transaction per block. Table III shows the blockchain coping with a doubling in speed of incoming data giving a delay of 4 seconds. In table IV the increased data flow leads to multiple transactions being written to the same block with no average decrease in speed indicating the scalability of the blockchain network for the requirements of our use case. In Table V, as the message volume increases, we see the failure of the web3 library[19] itself rather than Kaleido to cope with the transmission. The results show Kaleido blockchain easily copes with 10x and 100x our requirement.

TABLE II 0.1 msg/s WRITES

0.1 msg/s Results Analysis for Blockchain Writes			
Device Time Stamp	***Block ID***	***Kaleido Time Stamp***	***Delay***
ID : 1-2022 20:36:17	51541	08:36:22	5s
ID : 2-2022 20:36:27	51543	08:36:32	5s

TABLE III 0.2 msg/s WRITES

0.2 msg/s Results Analysis for Blockchain Writes			
Device Time Stamp	***Block ID***	***Kaleido Time Stamp***	***Delay***
ID : 1-2022 21:15:13	52008	09:15:17	4s
ID : 2-2022 21:15:18	52009	09:15:22	4s

TABLE IV 1 MSG/S WRITES

1 msg/s Results Analysis for Blockchain Writes			
Device Time Stamp	***Block ID***	***Kaleido Time Stamp***	***Delay***
ID : 1-2022 21:25:01	52126	09:25:07	6s
ID : 2-2022 21:25:02	52126	09:25:07	5s

TABLE V 10 msg/s WRITES

10 msg/s Results Analysis for Blockchain Writes			
Device Time Stamp	***Block ID***	***Kaleido Time Stamp***	***Delay***
ID : 29 2022 21:44:39	52361	09:44:42	3s
ID : 302022 21:44:49			

When benchmarked against the manual process which conservatively takes 160 seconds, the test results far exceeds the time taken to manually complete the task. In the proposed solution the data is shared with multiple members of the blockchain consortium almost instantaneously. Latency is not the only criterion that would convince supply chain members to adopt this solution, but the guarantee of data security that comes with the blockchain is hard to debate against. The relative simplicity of the modular architecture which is mostly cloud-based makes it easy for any entity to adopt it for a trial period with little capital expenditure up front.

V. CONCLUSION AND FUTURE WORK

This paper has presented a blockchain-based traceability framework for product delivery where it also integrates with different smart devices at various points in that delivery chain. The main outcome of this paper is to simulate how a refrigerated product could be traced from the supplier, along the transport chain, and in the factory gate. It illustrates that data integrity is maintained, traceability information can be easily exchanged (and trusted) and business process automation can occur using smart contracts. The modular architecture is designed to address the common failures of complex IoT projects by moving towards a platform-type approach. The performance of the Kaleido blockchain exceeds the use case requirement.

Integrations of cloud, IoT and Blockchain are becoming more common, especially in use-cases involving supply-chain traceability. This will also offer more opportunities to

automate business processes that relied heavily on pen and paper in the past. This paper proposes an Enterprise-grade blockchain-as-a-service, allowing architects to focus on the business process and the core application integrations (OSS, MQTT and Blockchain libraries). The design considers both field and factory data sources and uses Node-Red low-code platform to glue the various elements together. Node.js provides the wrapper for the function calls to the Blockchain layer - this server also interacts with the Application layer. The Node.js server abstracts the complexity of the Blockchain interactions from the Application Layer, without losing any of the requirements around scalability, security and data privacy. The design proposed in this paper has the following advantages:

1) Improving scalability by storing only key data on the private Ethereum Blockchain.
2) Design is modular in nature so any one element can be replaced
3) Speed of access of the data to all parties in the consortium.
4) Automatic order placement through Smart Contracts having access to IoT data

Compared to the current paper-based process and standard client-server applications the data can take minutes if not hours to scan by hand and email it to another party - by which time the data on the paper may have changed again, or worse, the recipient has doubts as to the authenticity of the document. The shortcomings in the Blockchain-based solution are the technical knowledge gaps that may exist with each member of the consortia. Network failures could also lead to some data loss, though modifying MQTT's Quality of Service parameter (QoS=2 guarantees receipt, though it does increase network traffic considerably) can add resilience for network outages.

Other technologies like IPFS (Interplanetary File Storage) could be introduced easily via hosted provider features which allow for image storage in off-chain elements but where the file hash is stored on-chain [17], [18]. This option may be requested where the image of the delivery docket is still required.

This solution has been implemented in a test lab. We have ignored elements of the real solution such as Wi-Fi connectivity at the weigh-bridge site. Future solutions might incorporate eSim into RaspberryPi to facilitate remote communications. Local application servers in Module B could be moved to the cloud also, to assess the impact on performance. Future work would look at the levels of service that can be set using MQTT QoS to optimise the performance. Any future implementation would consider the addition of a suitable browser-based or mobile app solution to allow for easy interaction with the users as they query the data stored on the blockchain. There are many frameworks available in the blockchain space that accelerate this portion of the development process.

ACKNOWLEDGMENT

This research was supported by Skillnet Ireland which is funded by Department of Further and Higher Education, Research, Innovation and Science. This work is also supported by funding from ThermoKing Europe.

REFERENCES

[1] European Commission, "EU legislation to control F_gases", [Online]. Available : https://ec.europa.eu/clima/eu-action/fluorinated-greenhouse-gases/eu-legislation-control-f-gases_en [Accessed: Aug 5 2022].

[2] Coolin Post, "Europe proposes 6-years jail for F-gas breaches", [Online]. Available : **https://www.coolingpost.com/world-news/europe-proposes-6-years-jail-for-f-gas-breaches/ [**Accessed: Aug 5, 2022]

[3] L. Atzori, A. Iera, and G. Morabito. "The Internet of Things: A survey" *Computer Networks*, 54(15):2787–2805, 2010

[4] A. Arena, A. Bianchini, P. Perazzo, C. Vallati and G. Dini, "BRUSCHETTA: An IoT Blockchain-Based Framework for Certifying Extra Virgin Olive Oil Supply Chain," *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2019, pp. 173-179, doi: 10.1109/SMARTCOMP.2019.00049.

[5] Kaleido Blockchain Corporation, https://www.kaleido.io/

[6] P. Cui, J. Dixon, U. Guin and D. Dimase, "A Blockchain-Based Framework for Supply Chain Provenance," in *IEEE Access*, vol. 7, pp. 157113-157125, 2019, doi: 10.1109/ACCESS.2019.2949951.

[7] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things," *2017 International Conference on Service Systems and Service Management*, 2017, pp. 1-6, doi: 10.1109/ICSSSM.2017.7996119.

[8] H. Cui, Z. Chen, Y. Xi, H. Chen and J. Hao, "IoT Data Management and Lineage Traceability: A Blockchain-based Solution," *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, 2019, pp. 239-244, doi: 10.1109/ICCChinaW.2019.8849969.

[9] N. Teslya and I. Ryabchikov, "Blockchain Platforms Overview for Industrial IoT Purposes," *2018 22nd Conference of Open Innovations Association (FRUCT)*, 2018, pp. 250-256, doi: 10.23919/FRUCT.2018.8468276.

[10] Wikipedia, "Function as a Service – Cloud-based computing services" [Online] https://en.wikipedia.org/wiki/Function_as_a_service [Accessed : Aug 3 2022].

[11] J. Polge, R. Jemery and Y. Le Traon, "Permissioned blockchain frameworks in the industry: A comparison", www.sciencedirect.com/science/article/pii/S2405959520301909

[12] A. Alam, M. Shuaib, S. Garg, "Blockciahin-based Initiatives: Current state and common challenges – what solutions exist", www.sciencedirect.com/science/article/pii/S138912862100373X

[13] CertiK, "Blockchain Trilemma", [Online]. Available : https://medium.com/certik/the-blockchain-trilemma-decentralized-scalable-and-secure-e9d8c41a87b3 [Accessed : Aug 1 2022]

[14] Proof-of-Authority Chains, OpenEthereum Documentation, https://openethereum.github.io/Proof-of-Authority-Chains [Accessed : July 21 2022]

[15] Connecting with Truffle, Kaleido API Reference, https://docs.kaleido.io/developers/smart-contracts/truffle/ [Accessed : July 20 2022]

[16] Nodemon, npm package, https://www.npmjs.com/package/nodemon [Accessed : July 20 2022]

[17] "IPFS Storage The Easy Way", https://www.kaleido.io/blockchain-platform/ipfs-file-store [Accessed : Aug 2 2022]

[18] K. Azbeg, O. Ouchetto, & S. Andaloussi, Said. (2022). BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. Egyptian Informatics Journal. 23. 10.1016/j.eij.2022.02.004

[19] Spalladino, "Failed to check for transaction receipt", https://github.com/ChainSafe/web3.js/issues/2213