Decentralized Maintenance Event Documentation with Hyperledger Fabric

Clemens Wickboldt[1], Christoph Meise[2], and Natalia Kliewer[3]

[1,2,3] Freie Universität Berlin, Chair of Information Systems
clemens.wickboldt@fu-berlin.de, christoph.meise@fu-berlin.de,
natalia.kliewer@fu-berlin.de

Abstract. Due to the transparency and redundancy of blockchain, previously required intermediaries become obsolete, which enables the technology to align and digitize business processes in sectors with strong competition and low trust. On the example of the aviation industry, this contribution considers the application of blockchain in supply chains. Complete life cycle documentation is mandatory for safety-related aircraft parts. This work presents and evaluates an IT artefact storing the information of maintenance event certificates in a Hyperledger Fabric blockchain. Our research responds to the calls for practical applications in the blockchain research field. An existing proof of concept is advanced towards a more naturalistic environment by further decentralizing the system. The results suggest a growth in performance in regard to transaction throughput, latency and memory usage by distributing the system on different physical machines. Projectable patterns are identified that can be applied to a broad solution space in similar industry sectors.

Keywords: Maintenance Event Documentation, Hyperledger Fabric, Blockchain, Aviation Industry, Supply Chain

15th International Conference on Wirtschaftsinformatik,
March 08-11, 2020, Potsdam, Germany

https://doi.org/10.30844/wi_2020_b2-wickboldt

1 Introduction

Blockchain technology as a combination of existing technologies including encryption, decentralized data management and consensus building was introduced in 2008 [1]. As the underlying core mechanism for Bitcoin, it was first implemented in 2009. Blockchain is characterized by its decentralization, anonymity, auditability, persistency and therefore its ability to save costs and improve efficiency [2]. Today, blockchain technology is broadly considered a promising technology that can be applied to manifold applications beyond cryptocurrencies [2, 3]. Due to the transparency and redundancy of blockchain, previously required intermediaries become obsolete. This enables the technology to be used in several business areas. Individual value chains migrate through digitization to a shared business process. Encryption within the blockchain contributes to the fact that this consolidation is also possible in processes in which confidential data is used [4]. There is already a great demand for blockchain solutions in a variety of industries, such as healthcare, fintech, real estate, government [5] and supply chain [6]. Within the supply chain industry, data is currently sent by trustworthy third parties. [7] argue that existing services lack fundamental functionality including the tracking of information flows. Blockchain addresses this lack through cryptographic chaining of transactions and decentralized storage. [7] point out that the use of Blockchain accelerates the integration of digital supply chains.

Supply chain, especially in the aviation industry is characterized by a vast number of network participants on a global scale. The International Air Transport Association (IATA) states that component maintenance costs account for the second largest share (24%) of airlines' direct maintenance costs [8]. Due to the fact that competitors and regulators are distributed globally, the aviation industry is particularly suitable for digitizing processes with the help of blockchain. The safety of aircrafts and their components plays a central role in the aviation industry. A complete life cycle documentation for safety-relevant parts is required which is essential for further usage.

The current analog process for documentation of maintenance events on paper certificates is error-prone and leads to scrapping of high-priced aircraft parts when in doubt about the maintenance history. Due to the intense competitive situation and the large number of stakeholders involved, the implementation of a central digital solution has not been successful yet. Existing research lead to a proposition of a decentralized solution based on blockchain technology [9]. The implementation of a Proof of Concept on Hyperledger Fabric is described in [10]. The artefact is benchmarked and evaluated in a laboratory environment in [11]. This article takes up and continues this existing research.

Within this scope, the contribution of our research is threefold: First, by advancing a blockchain-based IT artefact towards a naturalistic environment, this research responds to the calls for applications in the blockchain research field to a great extent. Second, we find projectible patterns [12] in the evaluation of the instantiation that may be used to solve problems in other sectors. Third, the efficacy and efficiency of the existing proof of concept is proven in a more naturalistic environment. This

https://doi.org/10.30844/wi_2020_b2-wickboldt

contribution builds on existing literature on a proof of concept and extends the evaluation according to the Framework for Evaluation in Design Science Research (FEDS) by [13]. Building upon the work of [9], [10] and [11], the present paper forms the fourth evaluation episode in the FEDS evaluation process by iteratively extending it to a more naturalistic environment which requires a stable throughput on multiple physical machines while supporting an increasing number of users.

We rely on the design science research approach [14] to structure the remainder of this paper as follows: In section 2 we give a systematic overview of the problem area and existing solutions. Furthermore, we provide insights into the business process for maintenance event documentation in the aviation industry, as the underlying business environment for the development and evaluation of a (Blockchain-based Certification Storage System) BCSS. The third section presents the research methods used. Section 4 describes the design research process as an evolution of BCSS towards the resulting artefact. Design decisions made are permanently questioned and subjected to the requirements and restrictions of the business process [15] in order to ultimately respect the laws of the existing environment [16]. The results of the evaluation of the artefact are presented in detail in Section 5. Section 6 discusses the results and provides practical implications. A conclusion and outlook are given in Section 7.

2 Problem Description and State of the Art

The research field around Blockchain technology calls for "blockchain applications" [2]. Due to their globalized nature, business processes in supply chain and logistics are particularly in need for digitization. [17] state that network participants agree to share company-owned insights if they can be certain that this is not shared with the competitors. Hence, there is a trust issue which must be overcome which is impossible when relying on a centralized solution. There are early applications of blockchain technology which currently transforms the industry: Blockchain in maritime logistics has already been investigated and mapping of logistics processes via blockchain has been considered [18]. Other examples are the support of supply chains of agricultural products in the Asian region with blockchain with regard to traceability [19, 20]. In the energy sector, blockchain-based supply chain solutions are also considered more closely [21, 22]. Moreover, there are concepts regarding the supply chain of the aviation industry [23].

According to [2] blockchains fall into one of three categories: public, private and consortium. This paper focuses on the consortium and private blockchain. They are characterized by the fact that network participants, in contrast to public blockchains, are not pseudonymized. In addition, they are permissioned, efficient and the consensus algorithm is either executed partially decentralized through the nodes of selected organizations or centralized by a single instance. A recognized and common framework for private and consortium blockchain is Hyperledger Fabric. It allows to address specific characteristics of the underlying business process in a so-called *Endorsement* process.

2.1 Hyperledger Fabric

Hyperledger Fabric is an open source blockchain framework implementation under the umbrella of the Linux Foundation. It is characterized primarily by its modular architecture, as well as its extensibility. Unlike public blockchains, which allow unknown entities to participate in the network, Hyperledger Fabric networks require participants to register with a trusted Membership Service Provider (MSP). Furthermore, Fabric is the first system to enable the development of distributed applications using General Purpose Languages without an existing technical dependency on a cryptocurrency [3]. The framework is characterized by an Execute-Order-Validate Blockchain architecture that differs from the regular Order-Execute design. The implementation of business processes is realized with the help of Smart Contracts or the Chaincode. A Fabric blockchain consists of a network of nodes, which get their identity from a *Membership Service Provider* (MSP). According to [3], a node in a fabric network can have the following roles: *Clients* create transaction proposals for execution, orchestrate execution, and send transactions over the network to the Orderers. *Peers* execute and validate transaction proposals. All peers manage the blockchain ledger, that is, the entire transaction history and the current status of the ledger. Only a subset of the peers execute transaction proposals, which are called endorsers. *Orderer Service Nodes* (OSNs) form the Ordering Service and determine the sequence of the transactions that are attached to the ledger.

2.2 Use Case: Documentation of Maintenance Events in the Aviation Industry

The Maintenance Repair and Overhaul (MRO) industry in the aviation industry consists of a number of interest groups which are part of the aircraft maintenance supply chain. Stakeholders include mechanics, original equipment manufacturers, MRO providers, airlines, traders and government agencies. Each stakeholder has different roles and has access to proprietary information within the industry. Each party must follow individual processes and release different data to document maintenance events. Aircraft spare parts are distinguishable into safety-related and non-safety-related. Safety-related parts include turbines, landing gear and control components. Non-safety related parts are for example seats and other parts of the aircraft interior.

The focus of this work lies on safety-relevant parts whose technical condition significantly influences flight safety. These parts require complete lifecycle documentation, also known as back-to-birth (BtB) documentation in order allow continued use and trade. Airlines often outsource maintenance of these parts and enter into contracts with MRO providers to ensure safe, fast and efficient maintenance [9]. The documentation process carried out with the help of certificates which are issued in paper form and physically kept next to the spare part itself. Each maintenance or trade event which requires documentation, another document is created and kept together with the rest of the documents. This method is highly susceptible to failure, as loss or damage to at least one of these documents results in immediate devaluation of the aircraft part. Without complete documentation, it is no longer allowed to trade

https://doi.org/10.30844/wi_2020_b2-wickboldt

the parts, or to build in an airplane. These requirements are mandated by the respective aviation authorities such as the European Union Aviation Safety Agency [24] in Europe and the Federal Aviation Administration [25] in the USA. Failure of documentation results into scrapping of the parts which in turn leads to huge losses for the airlines, as the value of these parts, according to industry experts, take on average a six-figure amount. Furthermore, industry experts state that about 50% of all traded life limited parts have an erroneous certificate documentation and must be scrapped.

In order to support and represent the interest of airlines, the organization IATA coordinates and standardizes many aspects of airline activities [26]. The aim of IATA is to represent, guide and serve the air transportation industry. To this end, the organization develops global and commercial standards on which the aviation industry is built. One of these proposed processes is the Ideal Component Repair Cycle, which describes the cost-efficient course of repair of an aircraft part [8]. The goal of the process is to minimize turnaround time (TAT). This is defined as the elapsed time between removal of a component from the aircraft and return to the operator with the status functional capability. The amount of the TAT determines the cost of a repair, since the corresponding part indirectly incurs costs during this time due to the required replacement. In each step of the process, a certificate is issued. The following types of certificates have been identified by reviewing example certificates provided by domain experts: Parts List Report, Parts Profile Report, Parts Usage Report, Record of Inspection, Material Certificate, Storage Report, Bill of Sale, Generic Report and Repair Report. These certificates are only a subset of the actual existing certificate types, but they cover the component life cycle adequately. The approach presented here was validated by means of a truthful documentation of an aircraft part from practice.

2.3 Requirements and Limitations of the Business Process

An artefact for the digital documentation of maintenance events must meet certain requirements so that it can be used in practice. After consulting domain experts,[10] suggest the following metrics that are necessary for a successful implementation:

1. “Sales transactions must be executed immediately, others should be executed within 10 minutes. Confirmation of cycles per hour is only necessary at certain times; the speed of those queries is of secondary importance.”
2. “The system needs to be capable of handling a throughput of 5 Mio. transactions per year or ~10 transactions per minute or ~0.16 transactions per second (tps).”

During evaluation of the existing proof of concept, the following additional requirements had been identified in discussion with domain experts:

3. The system needs to be able to map the stakeholders in the aviation industry in the form of organizations on independent physical machines.
4. The performance must remain stable within the expected number of users.

5. The system must support a technical administrator role that is superordinate to other organizations.

As requirement 1 and 2 are addressed in [11], requirement 3,4 and 5 are specifically addressed in this article.

3 Research Method

For structuring the creation, evaluation and presentation of the artefact, we rely on the Design Science Research (DSR) framework [15]. The viable IT artefact is a proof a concept prototype based on the blockchain technology that enables the documentation of maintenance events in the aviation industry. The problem relevance is given, since a central digital solution has not been successful yet. Requirements as a basis for evaluation are developed conjointly with domain experts in the MRO industry which ensures relevance of our work. We ensure research rigor by using well-established frameworks [13-15]. To get to the research contributions, our work is based on existing literature about blockchain applications [1, 3, 27]. Moreover, it is important that every artefact instantiation is scientifically evaluated and researchers should rigorously demonstrate quality, utility and efficacy of evaluation methods [15, 28]. In order to achieve this, this evaluation is based on the Framework for Evaluation in Design Science Research. We address technology-oriented audiences by providing insight into the technology choices, such as the blockchain framework, Docker Swarm and Apache Kafka. To address management-oriented audiences, we describe the underlying business process in the MRO industry.

The followed trajectory in a DSR project depends on respective needs and available resources and therefore there are different evaluation strategies that can be followed. [13] state, that the functional purpose of evaluation can be either formative or summative. Moreover, paradigm of evaluation can be either naturalistic or artificial. Each of the possible strategies operate as a progression within these two dimensions from the origin towards a final evaluation. For this artefact, the greatest design risk is the underlying blockchain technology and is technically oriented. Furthermore, evaluations in a real system with the actual companies and stakeholders in the aviation industry are very expensive. The quality of the artefact itself is to be evaluated and is the basis for a possible cooperation along the supply chain. Due to these circumstances, the Technical Risk & Efficacy evaluation strategy is chosen. The main aspect of this strategy is the early use of formative evaluations, in order to be able to influence design decisions and detect difficulties as early as possible and to therefore reduce costs and risks [13]. This contribution aims to provide a formative evaluation in order to improve the quality of the BCSS. Building upon the work of [9], [10] and [11] the present contribution forms the fourth evaluation episode in the FEDS evaluation process. In this context, the goal is to increase the technical depth by addressing additional business needs.

The evaluation is realized by performing benchmarks. This paper is based on the work of [27], who provide insights into performance measurement and optimization leverage in Hyperledger Fabric Blockchain systems using Hyperledger Caliper, a benchmark tool for Hyperledger Fabric which was already used in [11] for a simple network configuration. This network is now enhanced to meet the additional requirements 3-5.

4 Artefact Description

The BCSS consists of three modules that interact with each other. The communication between the modules takes place using the JSON data format. The system implements four methods for performing transactions to write and read maintenance event certificates. These methods receive parameters about the status of the spare part. Function F1 *writeAsset* creates a new aircraft part with information such as serial number and part owner. An aircraft part is a digital asset that is stored in the Blockchain. Subsequently, F2 *writeCert* is used to append a certificate to the history of an asset. As a result of a variety of different certificates in the real world, the information is abstracted with a generic detail field. The stored data of these methods can be read with functions F3 *queryAsset* and F4 *queryCert*, in order to receive the current state of the digital asset or the complete certificate history.

The sequence diagram shown in Figure 1 shows the information flow between the individual modules when creating an aircraft part in the blockchain. It shows the process from the input of the user in the frontend to the feedback from the blockchain. After the registered user has filled out and confirmed a form for creating a new aircraft part using the user interface, the entered information is transferred to the *createCertificate* method implemented in the backend in form of a JSON string. A transaction request is then created from the transferred data. This contains additional information, such as the Chaincode id with the desired version and the identification of the client. The transaction request is then sent to one or, if necessary, several peers. The method for creating a certificate is now executed in the Chaincode. If this has been successfully completed, the peers execute this transaction. The current state of the database is considered to generate transaction results. This result includes a write and read set, as well as a response value. The transaction is then verified by at least one endorsing peer. For example, the endorsing policy checks whether the sender has sufficient rights to send the transaction. If an existing authorization can be verified, a successful response to the backend, and distributed it to all peers using the Orderer. The Ordering Service takes care of the chronological order of the transactions within the channels. It creates the blocks with the transactions for each channel. The transactions in the blocks are also marked for validity. Each peer now adds the new block to its chain and updates the changes to the valid transactions in its state database (here CouchDB). An event about the completion and writing of the transaction is then created and the result sent to the backend. This is then forwarded to the frontend and the user is presented with feedback on the success of the transaction via the user interface.

https://doi.org/10.30844/wi_2020_b2-wickboldt

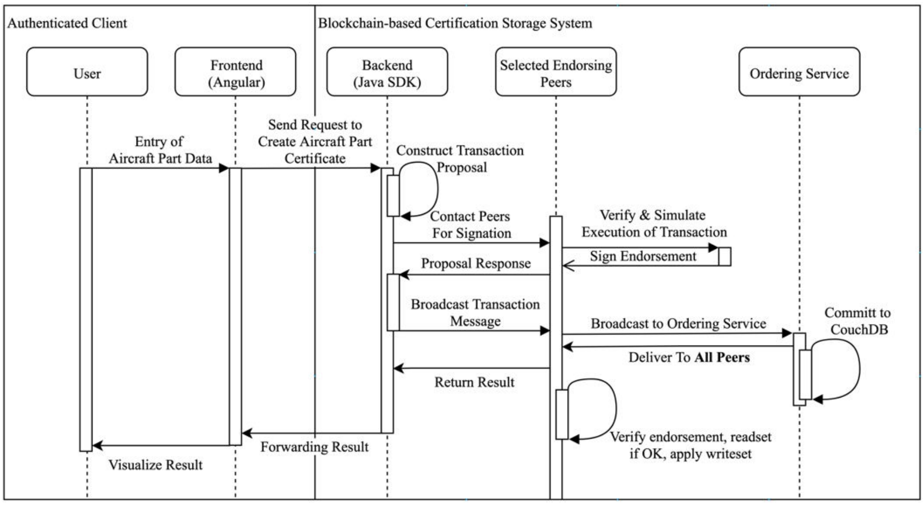


Figure 1. Sequence Diagram on the example of creating a certificate

4.1 Decentralized Architecture

A crucial aspect of blockchain is decentralization. Following the Technical Risk & Efficacy evaluation strategy, the artefact is first developed and evaluated in an artificial laboratory environment and afterwards iteratively extended by more naturalistic requirements. For the purpose of simplicity, the artefact has been evaluated formative and then artificially on a single machine [11]. With this next evaluation step, the BCSS is now distributed over several machines. Hyperledger Fabric offers the possibility to operate the Ordering Service in a Kafka mode. A setup based on Apache Kafka will be used to order the transactions from the peers. This instance of Apache Kafka offers consistency despite node crashes and scalable publish-subscribe messaging [3]. In Fabric, each channel is assigned to a separate single-partition topic in Kafka. The OSNs function as proxies between Kafka and peers. Moreover, they are independent of Kafka and could run on other physical machines. The Kafka setup consists of multiple nodes in a Kafka Cluster and a ZooKeeper ensemble. The minimum number of nodes in the Kafka Cluster is four, in order to be fault tolerant. If one of the brokers crashes, all existing channels are readable and writable, and it is possible to create new channels. The number of nodes in the ZooKeeper ensemble is either three, five, or seven. In order to avoid split-brain scenarios, it must be an odd number, and it has to be larger than one to avoid having a single point of failure.

In order to test the effects of decentralization on performance, the complexity of the architecture is iteratively increased. The first step is to run the Kafka cluster on a single machine and compare it with the previous solo mode. The network is then divided between two separate physical machines. The individual components are each operated in docker containers. Docker Swarm is used so that they can still communicate with each other. A virtual network is set up by the first machine M1, the

Swarm Manager. Additional machines can join this network with a unique token and the IP address of M1. The containers can then be operated within this network and can be distributed via the hostnames of the machines. The communication between the containers is done within this network as if it were on a single physical machine. As an alternative to this procedure, Kubernetes can be used, the containers are then orchestrated in a master-slave architecture. Since this requires a further external dependency, which can influence the performance, the Docker Swarm structure was preferred for the sake of simplicity.

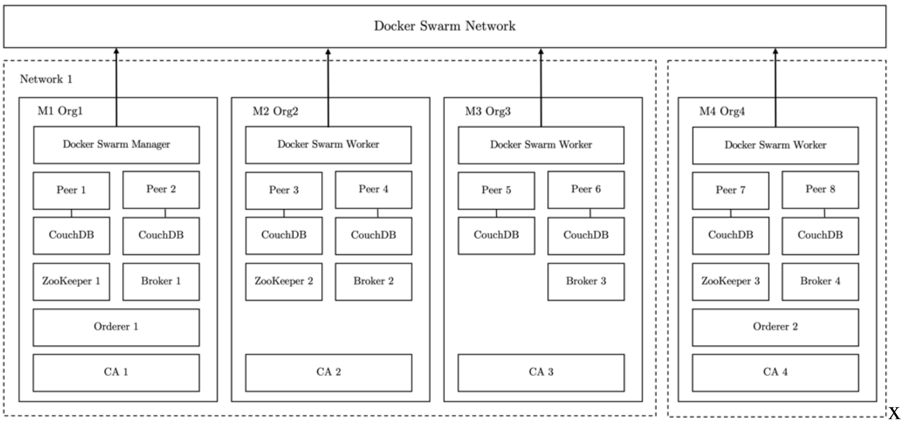


Figure 2. System Architecture

Figure 2 shows the technical architecture with four physical hosts, their respective components and their inter-communication: Two peers are operating on each machine. Machine M1, M2 and M4 provide ZooKeepers and every machine has a Broker. Furthermore, a CA is running on every machine. Due to the redundancy and distribution of the Kafka cluster and the CAs, this setup already largely corresponds to a realistic architecture. The fourth physical machine is outside the network. From a technical point of view, this structure corresponds to the requirements of reality and could be used in this form in a business application.

5 Evaluation

Utility, quality, and efficacy of the artefact [15] is demonstrated by ensuring that the artefact meets the business requirements. This is done via performance benchmarks with Hyperledger Caliper. Reports created by Caliper include the transaction throughput in second (tps), failed transaction due to timeouts, transaction latency and resource utilization. It is performed as follows: A transaction for a new aircraft part filled with realistic data is generated with F1. Afterwards, method F2 is used to create a certificate for a maintenance event that is attached to the history of the previously created asset. One transaction using F1 has a size of 160 bytes. The third and fourth operations are reading the system information (F3) and reading the certification information (F4).

5.1 Configuration Parameters and Experimental Setup

The benchmarks are conducted on four machines, while M1, M2 and M3 are in the same network and M4 is running on a separate network. With this, the architecture is more complex and is closer to conditions in a possible real-world application where the machines are completely independent of each other. Therefore, a benchmark with the external machine M4 may result to higher latencies as a result of the network separation. The machines M1, M2 and M3 have the following components: Intel(R) Xeon(R) Gold 6140 CPU @2.30GHz, 8 GB DDR4 RAM (ECC), 320 GB Serial Attached SCSI. M1 is running on Ubuntu 18.04 LTS while M2 and M3 are running on Debian GNU/Linux 8 (jessie). Again, this aspect serves to ensure that the system is as realistic and heterogeneous as possible. Moreover, Machine M4 has a different hardware configuration and consists of the following: two cores of an Intel(R) Core (TM) i7-3930K CPU @ 3.20GHz, 4GB DDR4 RAM and a 300GB HDD. Hyperledger Fabric is running in version 1.4 on all machines.

This benchmark provides insights about the performance of Fabric under various conditions and helps to understand how certain parameters affect the performance regarding the underlying business process of storing maintenance event certificates. Since the performance from a peer's perspective has already been studied thoroughly [3, 27], this paper focuses on the Orderer and network perspective. In all tests the endorsement policy is constantly "OR('Org2MSP.admin')" and requires a signature. This means that there is no additional computing power required to fulfill the endorsement policy and instead all requests that come from the administrator are fulfilled by default. There is only one channel in all setups and the peer database is CouchDB.

5.2 Setup Configurations

The tests for each setup are repeated with different configuration parameters. The configuration of the tested systems is shown in Table 1.

Table 1. Setup Configurations

Machines	*Orderer Mode*	*Orgs*	*Peers*	*Orderers*	*ZooKeepers*	*Kafkra Brokers*
1	Solo	2	4	M1	-	-
1	Kafka	2	4	M1	M1	M1
2	Kafka	2	4	M1	M1	M1
3	Kafka	3	6	M1, M2	M1, M2, M3	2 on M1, M2, M3
4	Kafka	4	8	M1, M4	M1, M2, M4	M1, M2, M3, M4

The number of transactions per test round is constantly 1000. The transaction arrival rate, also called send rate, stands for the number of Chaincode calls that Hyperledger Caliper sends to the system per second. In Hyperledger Fabric the Orderers control the number of messages batched into a block (block size). Moreover, the batch timeout is configured to be 2 seconds, which is the amount of time to wait before creating a batch. A benchmark is conducted for the transaction arrival rates t ∈20,40,50,60,80,100, the number of concurrent users 2^c, c ∈1,2,4,8,16 and the block size 2^b, b ∈1,2,4,8,16,32,64,128,256,512.

5.3 Benchmarking Results

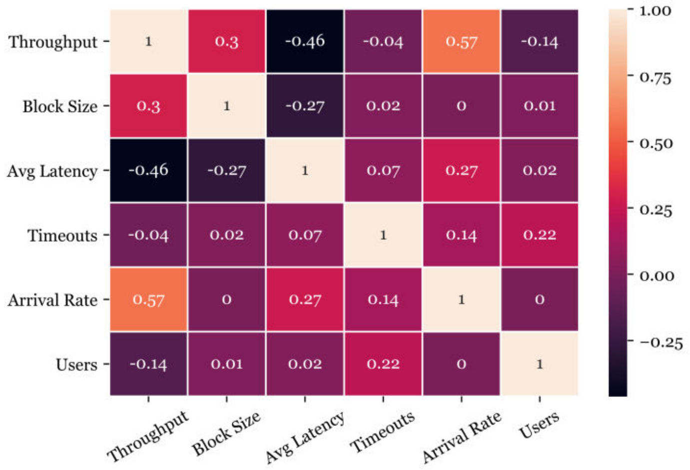
The results of the benchmarks are presented in the following. Throughput and latency results are shown along the parameter block size and number of concurrent users. The results of disc utilization are presented along block sizes.

Timeouts per Throughput and Transaction Arrival Rate. Timeouts multiply when the maximum system load is reached. For the machines used in this benchmark the average maximum for all tested configurations is at approximately 80 transactions per second.

Setup Performances. The transaction throughput is measured for different numbers of concurrent users. Write operations have a lower throughput than read operations and require more computing power. The setup on a single physical machine with the Solo Orderer has the lowest overall throughput with an average of 24.47 tps for read and 17.28 tps for write operations. The single host setup with the Kafka Orderer showed a significant performance growth with an average of 31.30 tps for read and 21.46 tps for write operations, performing 26.39% better on average. By distributing the system on two physical machines, the performance increases to an average of 41.65 tps for read operations and 34.12 tps for write operations. As a result, the mean throughput increased by 43.60% if a second physical machine is used. The results suggest that a distribution of the system on three machines is nearly identical, with a difference of only 0.59 tps on average. The performance increases slightly (15.4%) when the fourth machine is added to the system, which is a result of the faster CPU in M4.

The average memory usage is dependent on the block size and the system setup. The most memory was needed with the Solo configuration and a maximum memory usage of 170.48 MB for a block size of 512. The Kafka configuration on 2 physical hosts showed the lowest overall maximum memory usage with 95.28 MB with 2 transactions per block. For the distributed system with 3 and 4 physical machines, the minimum average memory usage is higher. All configurations reach the maximum transaction throughput with a block size of 128.

Parameter Configuration. Figure 3 shows a correlation heatmap for benchmarking parameter and results. The block size and throughput are positively (0.3) correlated to each other, indicating that a larger block size leads to a lower throughput. Average latency and throughput have a negative (-0.46) correlation, suggesting that a higher throughput increases the average latency. Arrival rate and throughput are also positively (0.57) correlated to each other. Users and throughput have a positive (0.22) correlation, suggesting that a higher number of users leads to a higher number of timeouts. Moreover, the number of concurrent users and throughput correlated negatively (-0.14) to each other. A higher number of users therefore indicates a lower throughput.



	Throughput	Block Size	Avg Latency	Timeouts	Arrival Rate	Users
Throughput	1	0.3	-0.46	-0.04	0.57	-0.14
Block Size	0.3	1	-0.27	0.02	0	0.01
Avg Latency	-0.46	-0.27	1	0.07	0.27	0.02
Timeouts	-0.04	0.02	0.07	1	0.14	0.22
Arrival Rate	0.57	0	0.27	0.14	1	0
Users	-0.14	0.01	0.02	0.22	0	1

Figure 3. Correlation Heatmap for Benchmarking Parameter and Results

Transaction Throughput and Failed Transactions. Write and read operations were tested with transaction send rates of 20, 40, 50, 60, 80 and 100 transactions per second. The systems performance is decreasing slightly with an increasing number of concurrently active clients. The largest decrease in performance was measured for a send rate of 100 read operations per second. The maximum throughput was 64.07 tps on average for 1 user and the minimum was at 49.23 tps for 16 concurrent users, a performance reduction of approximately 23.16%. In contrast, for a send rate of 20 read operations per second the throughput decreases by about 19.40%, which is equal to approximately 3.82 tps.

The number of concurrent users and the number of transaction timeouts have a positive correlation. For write operations with a high send rate of 80 and 100 tps, the number of timeouts increases significantly when testing with more than 16 and more than 8 concurrent users, respectively. Another benchmark with a block size of 128 with the configuration on 4 machines suggested that the maximum number of users for the test systems is 45, with an average fail rate over all send rates of more than 9.46%. Tests with 32 users showed an average fail rate of about 3.43%.

The results suggest that the average transaction latency decreases significantly until a block size of 32 for all transaction arrival rates. The maximum average latency of 76.95 ms is reached at a send rate of 80 and 1 transaction per block for writing. The minimum average latency of 1.11 ms is reached at a transaction arrival rate of 20 and a block size of 4 for reading.

6 Discussion

Evidence of the evaluation via performance benchmarks proves that the artefact meets the specified requirements. In order to ensure research rigor we compare our results with existing literature and our previous work [11]. [27] observe that the throughput is highly dependent on the number of CPU cores, which explains the better performance in throughput for distributed setups. This is coherent with our results. [27] suggest that the number of concurrent users correlates negatively with the performance, which could not be confirmed by the conducted benchmarks.

In our previous work, a setup on a single machine with the Orderer mode Solo was tested. Our current results are consistent with this previous work on several points: With a single machine and one orderer, a positive correlation (0.53) between the block size and throughput has been observed. Moreover, the benchmarks indicated a negative
(-0.84) correlation between the average latency and transaction throughput. These results could be confirmed by the conducted benchmarks within the scope of this contribution. Also, a stable performance is given to 16 concurrent users in a laboratory environment.

These results have been confirmed in a more naturalistic environment. The conducted benchmarks suggest that a higher number of physical machines, organizations and peers have no negative impact on performance. On the contrary, the throughput of distributed systems was significantly higher than single machine environments.

7 Conclusions

The outcome of this work includes the advancement of a blockchain-based IT artefact towards a more naturalistic environment and responds to the call for “blockchain applications” in [2]. We found projectible patterns [12] that researches can adapt to comparable issues in other sectors. In accordance with the Technical Risk & Efficacy evaluation strategy [13], the artefact is first designed in a simple laboratory environment, and then iteratively adds and examines reality requirements. The evaluation proves the efficacy and efficiency of the proof of concept in a naturalistic environment and extends the work of [9], [10] and [11]. The performance of Fabric in the context of maintenance event certificates stored digitally in a BCSS. Caliper was used to run the benchmarks. Various system architecture configurations as well as some crucial parameters were tested.

Within this article, five stages of evaluation were examined. First, a minimal and simplified setup on a single machine was tested. Afterwards the fault tolerant and high-performance Kafka service was introduced. The results indicate a great performance gain when using Kafka compared to the development configuration Solo. Since the underlying business process involves multiple independent stakeholders, a second machine was added to the system in the third evaluation stage. As a result, the transaction throughput increased by 43.60%. In this configuration most components were running on the first machine, which is a single point of failure. Therefore, a third physical host was introduced in the fourth step. In this setup, there is no single point of failure with all components distributed between the machines. There was no measurable performance difference to the prior configuration. In the evaluation stage, the impact of latency was investigated by adding a fourth machine in a different network. The fourth machine had faster hardware, which is why the performance increased slightly.

The requirements (1) and (2) have already been confirmed in prior research and the conducted benchmarks verify the results [11]. The requirement (3) was addressed by adding multiple physical machines to the system, which represent the different stakeholder in the aviation industry. Since the performance only increased or stagnated when adding more machines, a higher number of stakeholders is expected to not influence the systems performance negatively. Requirement (4) was addressed by testing the impact of the number of concurrent users on the system. The results suggest a negative impact on the transaction throughput and the relative number of timeouts for high send rates and high number of users. This requirement is fulfilled in consideration of the very low expected throughput and the moderate number of expected users. By showing that the Fabric components are interchangeable between the organizations, the requirement (5) was addressed. A machine with a central CA can be used to represent a technical administrator in the system.

The results indicate a significant increase in transaction throughput, with lesser timeouts using Apache Kafka. The same could be observed by distributing the system over several machines. Due to the overall higher performance of all machines in sum, the latency has also decreased. The network latency did not significantly reduce the performance of the system. The various configurations also found that a block size of 128 provides very good results in terms of latency, throughput and memory usage. Furthermore, the results regarding the influence of the number of users on existing research coincide [27]. Thus, an increasing number of users has a negative effect on the transaction rate. Since the expected transaction arrival rate in a real application is comparably low (2), this should not decrease the performance of the system in a naturalistic environment.

Following the Technical Risk & Efficacy evaluation strategy, future research should move the system to an even more naturalistic environment and evaluations should be done in a more summative way. This indicates including a larger circle of domain experts and testing in a production environment with different user roles. The current results indicate that the performance of the proposed system exceeds the requirements and that it can be used as a basis for real-world applications.

https://doi.org/10.30844/wi_2020_b2-wickboldt

The adoption of a BCSS in the aviation and similar industries leads to an accelerated integration of digital supply chain in the sense of [7]. The BCSS promotes the digitization of a joint business process that was previously not digitizable due to the confidentiality of the data. Thus it helps the integration of individual value chains to a shared business process in the sense of [4]. [29] argue that while the impact of using blockchain technologies is enormous, it will take decades for this impact to become visible in the economic and social infrastructure. The adaptation process is therefore incremental and continuous. The adaptation of BCSS to a naturalistic environment is a further step in this adaptation.

References

1. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. (2008)
2. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: Architecture, consensus, and future trends. In: Big Data (BigData Congress), 2017 IEEE International Congress on, pp. 557-564. IEEE, (2017)
3. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, pp. 30. ACM, (2018)
4. Tumasjan, A., Beutel, T.: Blockchain-Based Decentralized Business Models in the Sharing Economy: A Technology Adoption Perspective. Business Transformation Through Blockchain, pp. 77-120. Springer (2019)
5. Christidis, K., Devetsikiotis, M.: Blockchains and Smart Contracts for the Internet of Things. IEEE Access 4, 2292-2303 (2016)
6. Madhwal, Y., Panfilov, P.B.: Blockchain and Supply Chain Management: Aircrafts' Parts' Business Case. Annals of DAAAM & Proceedings 1051-1057 (2018)
7. Korpela, K., Hallikas, J., Dahlberg, T.: Digital supply chain transformation toward blockchain integration. In: proceedings of the 50th Hawaii international conference on system sciences. (2017)
8. IATA: Best Practices for Component Maintenance Cost Management. International Air Transport Association (2015)
9. Wickboldt, C., Kliewer, N.: Blockchain zur dezentralen Dokumentation von Werkstattereignissen in der Luftfahrtindustrie. HMD Praxis der Wirtschaftsinformatik 1-14 (2018)
10. Wickboldt, C., Kliewer, N.: Blockchain for Workshop Event Certificats - A Proof of Concept in the Aviation Industry. In: Proceedings of the 27th European Conference on Information Systems (ECIS). (2019)
11. Wickboldt, C.: Benchmarking a Blockchain-based Certification Storage System. Diskussionsbeiträge Fachbereich Wirtschaftswissenschaft, vol. 2019/5 : Information Systems. Freie Universität Berlin (2019)
12. Baskerville, R., Pries-Heje, J.: Design theory projectability. In: Working Conference on Information Systems and Organizations, pp. 219-232. Springer, (2014)

https://doi.org/10.30844/wi_2020_b2-wickboldt

13. Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. European Journal of Information Systems 25, 77-89 (2016)
14. Gregor, S., Hevner, A.R.: Positioning and presenting design science research for maximum impact. MIS quarterly 37, (2013)
15. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS quarterly 28, 75-105 (2004)
16. Simon, H.A.: The sciences of the artificial. MIT press (1996)
17. Dyer, J.H., Singh, H.: The relational view: Cooperative strategy and sources of interorganizational competitive advantage. Academy of management review 23, 660-679 (1998)
18. Stahlbock, R., Heilig, L., Voß, S.: Blockchain in der maritimen Logistik. HMD Praxis der Wirtschaftsinformatik 55, 1185-1203 (2018)
19. Kumar, M.V., Iyengar, N.: A framework for Blockchain technology in rice supply chain management. Adv. Sci. Technol. Lett 146, 125-130 (2017)
20. Tian, F.: An agri-food supply chain traceability system for China based on RFID & blockchain technology. In: Service Systems and Service Management (ICSSSM), 2016 13th International Conference on, pp. 1-6. IEEE, (2016)
21. Albrecht, S., Reichert, S., Schmid, J., Strüker, J., Neumann, D., Fridgen, G.: Dynamics of blockchain implementation-a case study from the energy sector. In: Proceedings of the 51st Hawaii International Conference on System Sciences. (2018)
22. Sikorski, J.J., Haughton, J., Kraft, M.: Blockchain technology in the chemical industry: Machine-to-machine electricity market. Applied Energy 195, 234-246 (2017)
23. Madhwal, Y., Panfilov, P.B.: Blockchain and Supply Chain Management: Aircrafts' Parts' Business Case. In: 28th DAAAM International Symposium on Intelligent Manufacturing and Automation. (2017)
24. EASA: Easa: Acceptable Means of Compliance (Amc) and Guidance Material (Gm). (2015)
25. FAA: Advisory Circular. (2015)
26. Doganis, R.: Flying off course: The economics of international airlines. Routledge (2013)
27. Thakkar, P., Nathan, S., Vishwanathan, B.: Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. arXiv preprint arXiv:1805.11390 (2018)
28. March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision support systems 15, 251-266 (1995)
29. Iansiti, M., Lakhani, K.R.: The truth about blockchain. Harvard Business Review 95, 118-127 (2017)